



# full circle

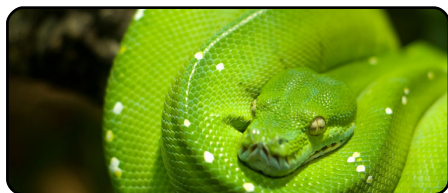
THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #28 - August 2009



# LINUX, APACHE, MYSQL & PHP SERVER (LAMP) PART 1

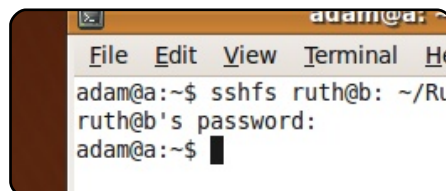




Program In Python - Pt2 p.07



LAMP - Part 1 p.12



Networking With SSHFS p.15

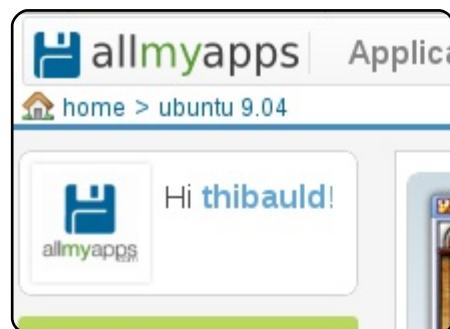


Fast Internet With Squid p.17



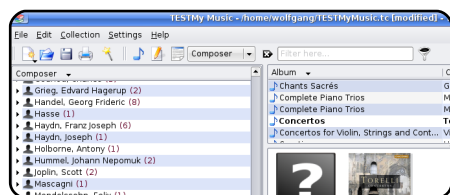
# full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY



My Opinion p.21

*Allmyapps* is a site where you can list the apps you use and, if need be, install them from the site with one click...



Review p.25

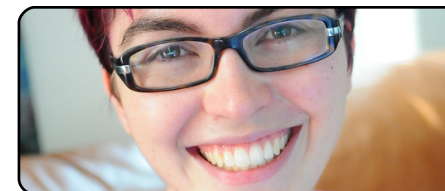


MOTU Interview p.27

This issue - Stephane Graber (stgraber) in the United States.



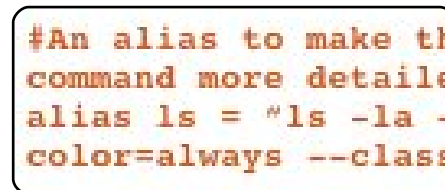
Letters p.28



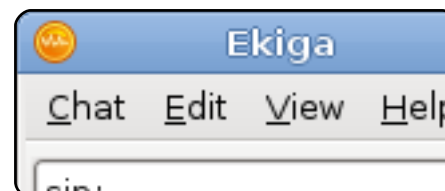
Ubuntu Women p.30



Ubuntu Games p.32



Command & Conquer p.05



Top 5 p.36



The articles contained in this magazine are released under the Creative Commons Attribution-Share Alike 3.0 Unported license. This means you can adapt, copy, distribute and transmit the articles but only under the following conditions: You must attribute the work to the original author in some way (at least a name, email or URL) and to this magazine by name ('full circle magazine') and the URL [www.fullcirclemagazine.org](http://www.fullcirclemagazine.org) (but not attribute the article(s) in any way that suggests that they endorse you or your use of the work). If you alter, transform, or build upon this work, you must distribute the resulting work under the same, similar or a compatible license.

**Full Circle magazine is entirely independent of Canonical, the sponsor of the Ubuntu projects, and the views and opinions in the magazine should in no way be assumed to have Canonical endorsement.**



## Welcome to another issue of Full Circle magazine.

**A**nd welcome to another new series! I promised you LAMP, and here it is: creating your own LAMP server. LAMP stands for Linux, Apache, MySQL and PHP (sometimes Python) and is the foundation upon which many an Internet server sits, so it is very widely used. In part one (this month) Richard Bosomworth discusses the installation and administration of a LAMP server and, next month, will touch on FTP and Firewalls, so stay tuned!

Also this month, Greg has a small correction to his Python Part 1 series, so make sure you check that out, and he goes on to talk more about variables.

In other news, just last week Karmic Koala (Ubuntu 9.10) reached Alpha 4, and August 31st brings the Ubuntu Developer Week (UDW):

*Ubuntu Developer Week is the perfect opportunity for you to get involved and get a closer look at what's happening behind the scenes. Make sure you mark the dates from Monday, 31st August 2009 to Friday, 4th September 2009 in your calendar! Just check out the timetable to learn more about which session is up next.*

The UDW timetable is at: <https://wiki.ubuntu.com/UbuntuDeveloperWeek>

Enjoy the issue, and keep in touch!

**All the best,**

Ronnie

Editor, Full Circle magazine

[ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org)

This magazine was created using :



## What is Ubuntu?

Ubuntu is a complete operating system that is perfect for laptops, desktops and servers. Whether at home, school or work Ubuntu contains all the applications you'll ever need including word processor, email application and web browser. Ubuntu is and always will be free of charge. You do not pay any licensing fees. You can download, use and share Ubuntu with your friends, family, school or business for absolutely nothing.

Once installed, your system is ready to use with a full set of productivity, internet, drawing and graphics applications, and games.

**TIP:** use the new 'contents' link to jump to the contents page from any other page!







## New Skype Beta For Linux



After over a year of silence from Skype's Linux developers since even a minor update to the Skype Linux client and over two years since the last major update was revealed, a new beta release of Skype for Linux has been made available. The new version 2.1 Beta uses the Skype SILK codec which should bring better quality audio, even on low bandwidth connections. It also now supports high quality video and can make use of PulseAudio. SMS messaging is now available, as is the ability to organise contacts into groups. The chat functionality has also been enhanced with better indications of typing, message editing and new emoticons.

The release notes detail the known issues with the new Beta version and details all the improvements made to the client. The software, which is under a proprietary license, is available to download for Ubuntu 8.04 and later versions, Debian Lenny, Fedora 9 and later, openSUSE 11, and as a generic, statically or dynamically linked binary.

**Source:** [www.h-online.com](http://www.h-online.com)

To download the new beta of Skype:  
<http://www.skype.com/download/skype/linux/>



## Microsoft Acknowledges Linux As Desktop Competitor

Microsoft listed Linux distributors Canonical and Red Hat among competing commercial software suppliers in its annual report to the US Securities and Exchange Commission.

The report said the Linux operating system had gained "some acceptance" as PC makers look to reduce costs and more people buy netbook PCs that mainly run Linux.

Windows Vista has proven to be too large and too costly for netbooks, which has driven adoption of Linux.

Microsoft's Windows 7, due for release in October, is to have a special light version optimised for use on netbooks.

**Source:** [www.computerweekly.com](http://www.computerweekly.com)

## Linux From Scratch 6.5: The DIY Operating System Guide



The Linux From Scratch project has released version 6.5 of its step-by-step instructions on how to create your own Linux system from scratch. The 300 pages of instructions show how to obtain the source code components and then compile your own 32-bit Linux system. LFS 6.5 has been updated to cover building a system using Linux kernel version 2.6.30.2 with GCC 4.4.1 and version 2.10.1 of glibc. LFS requires you already have a Linux system running to do the compilation and assembly of your own DIY Linux. Other updates to 6.5 are listed in the What's new since the last release page of the book.

**Source:** [www.h-online.com](http://www.h-online.com)



# COMMAND & CONQUER

Written by Lucas Westermann

Recently a reader had requested that I cover the basic layout of a help or man (manual) page. For the benefit of this article, I will focus on the 'help' and 'man' pages for the “ping” command. The command:

**ping -h**

will display the help command for ping (the “-h” switch, along with the “--help” argument, are the defacto default). The help information will look something like what is shown in the box above right.

The first square brackets containing “-LRUbdnqrvVaA” is a list of possible switches that don't require arguments (mainly because they format output), and for an explanation as to what any of these switches does, it's required to check the man page as well. The next series of square brackets that show a switch and value combination (e.g. “-c count”) requires you to

substitute the “count” section with an actual value. The words are intended to give you an idea of what the switch does.

Ideally, the help page is intended as a quick reference, in case you're unsure what switch corresponds to the input you wish to supply. If, however, you're new to the command and don't know what half of the switches do, it's best to read through the man page, since it offers an explanation of each and every switch, as well as possible uses, help website, etc. There are some help pages that offer descriptions of arguments and switches, as

```
Usage: ping [-LRUbdnqrvVaA] [-c count] [-i interval] [-w deadline]
          [-p pattern] [-s packetsize] [-t ttl] [-I interface or address]
          [-M mtu discovery hint] [-S sndbuf]
          [-T timestamp option] [-Q tos] [hop1 ...] destination
```

**count** - number of ping tries  
**interval** - time between pings  
**deadline** - maximum time to wait for reply  
**pattern** - allows you to fill the ping with a pattern of bytes  
**packetsize** - defines the size of the packet to send  
**ttl** - sets the IP Time to Live  
**interface or address** - Set the source IP or device  
**mtu discovery hint** - "do", "want" or "dont" are options  
**sndbuf** - defines the size of the send buffer  
**timestamp option** - allows you to set special IP timestamp options  
**tos** - sets the Quality of Service option (either decimal or hex value accepted)  
**hop1...** - can be a list of destinations

well as more information, but that isn't always the case.

To view the man page, use this command:

**man ping**

The synopsis section displays something similar to the help command, but it is followed by a description section, which

explains what the command is intended to do, or intended to be used for. The options section lists all the switches, and a short description of what each does. To navigate through the rest of the manpage, use the up and down arrows, or page up and page down. The text on the very bottom looks like this: “Manual page ping(8) line 21/356 15%”. It tells you

## COMMAND & CONQUER

that you are viewing the manual page for ping, and are at line 21 out of 356, or 15% of the way through the document. Once you're done viewing the rest of the options section, you come upon a section named "ICMP Packet Details", which defines what they are. There are a few more definitions before you reach the bug heading, but each one is fairly well explained, so I will not explain each one. The bugs listed are currently open (known bugs that haven't been fixed in that version), and the See Also: section offers a few other commands to look at that will be useful in combination with the command you are currently reading up on. The history section is straight-forward enough, while security and availability tells you a little more about the command.

Not all man pages are laid out in the same manner, but they follow the same conventions, which are:

**Name**  
**Synopsis**  
**Description**

**Options**  
**Useful information pertinent to the command (definitions, explanations, etc.)**  
**Bugs**  
**See Also**  
**History**  
**Security (if applicable)**  
**Availability**

This is useful in case you want to look up a specific term from the help page, since you know exactly where to go. Also, if you ever write your own tool, or would like to add a man page to a script you wrote, you then know how to format it. Hopefully this article has helped you to get a better understanding of how manpages work, and how to make sense of the often confusing help pages.



**Lucas** has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: [lswest34@gmail.com](mailto:lswest34@gmail.com).

# ATLANTA LINUX FEST

**SOFTWARE FREEDOM DAY!**

DATE:  
**Saturday  
September 19, 2009**

WEBSITE:  
**atlantalinuxfest.org**

COST:  
**Free!**

ADDRESS:  
**IBM  
4111 Northside Pkwy  
Atlanta, GA 30327**

All lovers of Linux and Open Source Software are invited to Atlanta Linux Fest: the place to learn, make new friends, and have fun! 🐧

**Atlanta Linux Fest**  
- FOR MORE INFO LOG ONTO -  
**atlantalinuxfest.org**





# HOW-TO

Written by Greg Walters

## Program In Python - Part 2

### SEE ALSO:

FCM#27 - Python Part 1

### APPLICABLE TO:

ubuntu kubuntu xubuntu

### CATEGORIES:



### DEVICES:



### Correction To Part1

I received an email from David Turner who suggested that using the Tab-key for indentation of code is somewhat misleading as some editors may use more, or less, than four spaces per indent. This is correct. Many Python programmers (myself included) save time by setting the tab key in their editor to four spaces. The problem is, however, that someone else's editor may not have the same setting as yours, which could lead to ugly code and other problems. So, get into the habit of using spaces rather than the Tab-key.

In the last installment, we looked at a simple program using `raw_input` to get a response from the user, some simple variable types, and a simple loop using the "for" statement. In this installment, we will delve more into variables, and write a few more programs.

### LISTS

Let's look at another type of variable called lists. In other languages, a list would be considered an array. Going back to the analogy of shoe-boxes, an array (or list) would be a number of boxes all glued side-by-side holding like items. For example, we could store forks in one box, knives in another, and spoons in another. Let's look at a simple list. An easy one to picture would be a list of month names. We would code it like this...

```
months =
['Jan', 'Feb', 'Mar', 'Apr', 'May',
'Jun', 'Jul', 'Aug', 'Sep', 'Oc
```

```
t', 'Nov', 'Dec']
```

To create the list, we bracket all the values with square brackets ( '[' and ']' ). We have named our list 'months'. To use it, we would say something like `print months[0]` or `months[1]` (which would print 'Jan' or 'Feb'). Remember that we always count from zero. To find the length of the list, we can use:

```
print len(months)
```

which returns 12.

Another example of a list would be categories in a cookbook. For example...

```
categories = ['Main
dish', 'Meat', 'Fish', 'Soup', 'C
ookies']
```

Then `categories[0]` would be 'Main dish', and `categories[4]` would be 'Cookies'. Pretty simple again. I'm sure you can think of many things that you can use a list for.

Up to now, we have created a list using strings as the information. You can also create a list using integers. Looking back at our months list, we could create a list containing the number of days in each one:

```
DaysInMonth =
[31, 28, 31, 30, 31, 30, 31, 31, 30, 3
1, 30, 31]
```

If we were to print `DaysInMonth[1]` (for February) we would get back 28, which is an integer. Notice that I made the list name `DaysInMonth`. Just as easily, I could have used 'daysinmonth' or just 'X'... but that is not quite so easy to read. Good programming practices suggest (and this is subject to interpretation) that the variable names are easy to understand. We'll get into the whys of this later on. We'll play with lists some more in a little while.

Before we get to our next sample program, let's look at a few other things about Python.

## More on Strings

We briefly discussed strings in Part 1. Let's look at string a bit closer. A string is a series of characters. Not much more than that. In fact, you can look at a string as an array of characters. For example if we assign the string 'The time has come' to a variable named `strng`, and then wanted to know what the second character would be, we could type:

```
strng = 'The time has come'
print strng[1]
```

The result would be 'h'. Remember we always count from 0, so the first character would be [0], the second would be [1], the third would be [2], and so on. If we want to find the characters starting at position 4 and going through position 8, we could say:

```
print strng[4:8]
```

which returns 'time'. Like our for loop in part 1, the counting stops at 8, but does not return the 8th character, which would

be the space after 'time'.

We can find out how long our string is by using the `len()` function:

```
print len(strng)
```

which returns 17. If we want to find out where in our string the word 'time' is, we could use

```
pos = strng.find('time')
```

Now, the variable `pos` (short for position) contains 4, saying that 'time' starts at position 4 in our string. If we asked the `find` function to find a word or sequence that doesn't exist in the string like this:

```
pos = strng.find('apples')
```

the returned value in `pos` would be -1.

We can also get each separate word in the string by using the `split` command. We will split (or break) the string at each space character by using:

```
print strng.split(' ')
```

which returns a list containing

['The', 'time', 'has', 'come'].

This is very powerful stuff.

There are many other built-in string functions, which we'll be using later on.

## Literal Substitution

There is one other thing that I will introduce before we get to our next programming example. When we want to print something that includes literal text as well as variable text, we can use what's called Variable Substitution. To do this is rather simple. If we want to substitute a string, we use '%s' and then tell Python what to substitute. For example, to print a month from our list above, we can use:

```
print 'Month = %s' %
month[0]
```

This would print 'Month = Jan'. If we want to substitute an integer, we use '%d'. Look at the example below:

```
Months =
['Jan', 'Feb', 'Mar', 'Apr', 'May',
'Jun', 'Jul', 'Aug', 'Sep', 'Oct',
'Nov', 'Dec']
DaysInMonth =
[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
```

```
1, 30, 31]
for cnt in range(0, 12):
    print '%s has %d
days.' %
(Months[cnt], DaysInMonth[cnt])
```

The result from this code is:

```
Jan has 31 days.
Feb has 28 days.
Mar has 31 days.
Apr has 30 days.
May has 31 days.
Jun has 30 days.
Jul has 31 days.
Aug has 31 days.
Sep has 30 days.
Oct has 31 days.
Nov has 30 days.
Dec has 31 days.
```

Something important to understand here is the use of single quotes and double quotes. If you assign a variable to a string like this:

```
st = 'The time has come'
```

or like this:

```
st = "The time has come"
```

the result is the same. However, if you need to include a single quote in the string like this:



```
st = 'He said he's on his way'
```

you will get a syntax error. You need to assign it like this:

```
st = "He said he's on his way"
```

Think of it this way. To define a string, you must enclose it in some kind of quotes – one at the beginning, and one at the end – and they must match. If you need to mix quotes, use the outer quotes to be the ones that aren't in the string as above. You might ask, what if I need to define a string like “She said “Don't Worry””? In this case, you could define it this way:

```
st = 'She said "Don\'t Worry"'
```

Notice the backslash before the single quote in 'Don't'. This is called an escape character, and tells Python to print the (in this case) single-quote – without considering it as a string delimiter. Other escape character sequences (to show just a few) would be '\n' for new line, and '\t' for tab. We'll deal with these in later sample

code.

## Assignment verses Equate

We need to learn a few more things to be able to do our next example. First is the difference between assignment and equate. We've used the assignment many times in our samples. When we want to assign a value to a variable, we use the assignment operator or the '=' (equal sign):

```
variable = value
```

However, when we want to evaluate a variable to a value, we must use a comparison operator. Let's say we want to check to see if a variable is equal to a specific value. We would use the '==' (two equal signs):

```
variable == value
```

So, if we have a variable named loop and we want to see if it is equal to, say, 12, we would use:

```
if loop == 12:
```

Don't worry about the if and the colon shown in the example above yet. Just remember we have to use the double-equal sign to do evaluation.

## Comments

The next thing we need to discuss is comments. Comments are important for many things. Not only do they give you or someone else an idea of what you are trying to do, but when you come back to your code, say 6 months from now, you can be reminded of what you were trying to do. When you start writing many programs, this will become important. Comments also allow you to make Python ignore certain lines of code. To comment a line you use the '#' sign. For example:

```
# This is a comment
```

You can put comments anywhere on a code line, but remember when you do, Python will ignore anything after the '#'.

## If statements

Now we will return to the "if" statement we showed briefly above. When we want to make a decision based on values of things, we can use the if statement:

```
if loop == 12:
```

This will check the variable 'loop', and, if the value is 12, then we do whatever is in the indented block below. Many times this will be sufficient, but, what if we want to say If a variable is something, then do this, otherwise do that. In pseudo code you could say:

```
if x == y then
    do something
else
    do something else
```

and in Python we would say:

```
if x == y:
    do something
else:
    do something else
    more things to do
```

The main things to remember here are:

1. End the if or else statements

with a colon.

## 2. INDENT your code lines.

Assuming you have more than one thing to check, you can use the if/elif/else format. For example:

```
x = 5
if x == 1:
    print 'X is 1'
elif x < 6:
    print 'X is less than
6'
elif x < 10:
    print 'X is less than
10'
else:
    print 'X is 10 or
greater'
```

Notice that we are using the '<' operator to see if x is LESS THAN certain values - in this case 6 or 10. Other common comparison operators would be greater than '>', less than or equal to '<=', greater than or equal to '>=', and not equal '!=',

## While statements

Finally, we'll look at a simple example of the while statement. The while statement allows you to create

a loop doing a series of steps over and over, until a specific threshold has been reached. A simple example would be assigning a variable "loop" to 1. Then while the loop variable is less than or equal to 10, print the value of loop, add one to it and continue, until, when loop is greater than 10, quit:

```
loop = 1
while loop <= 10:
    print loop
    loop = loop + 1
```

run in a terminal would produce the following output:

```
1
2
3
4
5
6
7
8
9
10
```

This is exactly what we wanted to see. Fig.1 (above right) is a similar example that is a bit more complicated, but still simple.

```
loop = 1
while loop == 1:
    response = raw_input("Enter something or 'quit' to end => ")
    if response == 'quit':
        print 'quitting'
        loop = 0
    else:
        print 'You typed %s' % response
```

FIG. 1

In this example, we are combining the if statement, while loop, raw\_input statement, newline escape sequence, assignment operator, and comparison operator – all in one 8 line program.

Running this example would produce:

```
Enter something or 'quit' to
end
=> FROG
You typed FROG
Enter something or 'quit' to
end
=> bird
You typed bird
Enter something or 'quit' to
end
=> 42
You typed 42
Enter something or 'quit' to
end
=> QUIT
You typed QUIT
Enter something or 'quit' to
end
```

=> quit  
quitting

Notice that when we typed 'QUIT', the program did not stop. That's because we are evaluating the value of the response variable to 'quit' (response == 'quit'). 'QUIT' does NOT equal 'quit'.

One more quick example before we leave for this month. Let's say you want to check to see if a user is allowed to access your program. While this example is not the best way to do this task, it's a good way to show some things that we've already learned. Basically, we will ask the user for their name and a password, compare them with information that we coded inside the program, and then make a decision based on what we find. We will use two lists – one to hold the allowed users and

one to hold the passwords. Then we'll use `raw_input` to get the information from the user, and finally the `if/elif/else` statements to check and decide if the user is allowed. Remember, this is not the best way to do this. We'll examine other ways in later articles. Our code is shown in the box to the right.

Save this as 'password\_test.py' and run it with various inputs.

The only thing that we haven't discussed yet is in the list checking routine starting with 'if `username` in `users`:". What we are doing is checking to see if the user's name that was entered is in the list. If it is, we get the position of the user's name in the list `users`. Then we use `users.index(username)` to get the position in the `users` list so we can pull the password, stored at the same position in the `passwords` list. For example, John is at position 1 in the `users` list. His password, 'dog' is at position 1 of the `passwords` list. That way we can match the two. Should be

```
#-----  
#password_test.py  
#    example of if/else, lists, assignments,raw_input,  
#    comments and evaluations  
#-----  
# Assign the users and passwords  
users = ['Fred','John','Steve','Ann','Mary']  
passwords = ['access','dog','12345','kids','qwerty']  
#-----  
# Get username and password  
username = raw_input('Enter your username => ')  
pwd = raw_input('Enter your password => ')  
#-----  
# Check to see if user is in the list  
if username in users:  
    position = users.index(username) #Get the position in the list of the users  
    if pwd == passwords[position]: #Find the password at position  
        print 'Hi there, %s. Access granted.' % username  
    else:  
        print 'Password incorrect. Access denied.'  
else:  
    print "Sorry...I don't recognize you. Access denied."
```

pretty easy to understand at this point.

**That's enough for this month. Next time, we'll be learning about functions and modules. Until then, play with what you've already learned and have fun.**



**Greg Walters** is owner of *RainyDay Solutions, LLC*, a consulting company in Aurora, Colorado, and has been programming since 1972. He enjoys cooking, hiking, music, and spending time with his family.





# HOW-TO

Written by Richard Bosomworth

## LAMP - Part 1

### SEE ALSO:

N/A

### APPLICABLE TO:

ubuntu kubuntu xubuntu

### CATEGORIES:



### DEVICES:



**T**oday's internet offers us more than just flat pages of information to scroll through. We are presented with contact forms, shopping carts, media libraries, interactive gaming, and social-networking portals - 80% of which operate through something called a LAMP stack

at the back end (on the server). LAMP is an acronym of open-source server-side technologies and is the most widely used solution stack across both the internet and the in-house business enterprise.

The **LAMP** breakdown is as follows:

**Linux** - The server operating system

**Apache** - The Web server platform

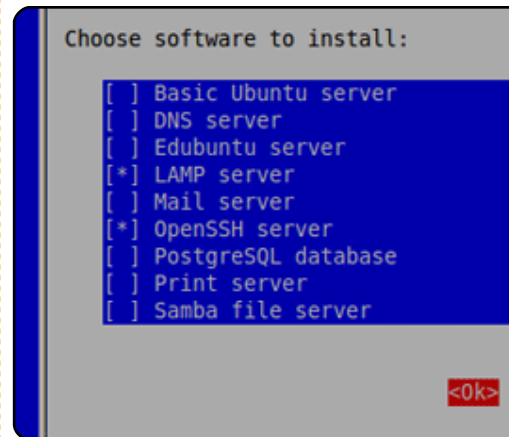
**MySQL** - The database

**PhpMyAdmin** - The dynamic scripting language, predominantly php although it can also be perl or python.

These big four were never designed with combined operation in mind. However, symbiotic suitability for Web and data delivery, coupled with cost-effective ubiquity of open-source components, evolved LAMP through the 1990s into the de-facto standard enabler for two-way data exchange and

information execution we all expect from our desktop browsers today.

This is the base LAMP stack installation procedure for Ubuntu Server 9.04 "Jaunty Jackalope".



Download the relevant server ISO for your system (32 or 64 bit, etc.), burn it to a CD, boot and and run through the install, selecting keyboard, location options + LAMP & Open SSH components. Once the installation is rolling, choose suitable user/account names & MySQL root passwords when asked.

All done? Good. Re-boot, log in, and update the system. From the CLI, type the following to update things. ('update' updates existing packages and 'upgrade' installs new versions -- if any are available.) Why not just upgrade first? Well, an update ensures initial reliable network connectivity and stable operational functionality of the system you have installed:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

As this is a server, we need to configure a static IP and bind it to the network adapter (invariably eth0). Use a text editor for this (I use vi. Search Google for the operational commands -- there are only a few you really need). At the CLI type:

```
sudo vi  
/etc/network/interfaces
```

Alter the DHCP entry to something like this (these are



my own IP numbers used as examples).

```
auto eth0
iface eth0 inet static
address 192.168.1.15
netmask 255.255.255.0
network 192.168.1.0
broadcast 192.168.1.255
gateway 192.168.1.1
```

With a static IP, you also may need to manually set a DNS entry in the resolv.conf file. This is quite easy. At the CLI type:

```
sudo vi /etc/resolv.conf
```

If the file is empty, don't worry, it's just creating a new one. However, if there are entries already listed, then leave them as they are. If empty, then enter details into the framework as follows (use your own domain name and chosen DNS IPs).

```
search domain.com
nameserver xxx.xxx.xxx.xxx
nameserver xxx.xxx.xxx.xxx
```

Restart your network services:

```
sudo /etc/init.d/networking
restart
```

## MySQL IP Bind

MySQL by default is connected to the localhost interface, which is not accessible from a non-local host. This can be changed by editing the /etc/mysql/my.cnf file. At the CLI type:

```
sudo vi /etc/mysql/my.cnf
```

You'll see the BIND entry. Just change it from localhost 127.0.0.1 to the IP number of the machine you will be accessing the database from, and save the file. Should you require access from multiple machines, comment out (#) the bind line, and either leave things open (not advised) or configure specific IP access to the DB later through phpMyAdmin instead. We shall be installing phpMyAdmin in the next section.

## Apache

To test Apache, type `http://<your ip number>` into a browser. You should see "It Works". If not, re-start Apache and try again. To re-start

Apache, at the CLI type:

```
sudo /etc/init.d/apache2
restart
```

**TIP:** When Apache restarts under certain circumstances you may see the following message:

```
apache2: Could not reliably
determine the server's fully
qualified domain name, using
127.0.0.1 for ServerName
```

Sites will still load, however. To fix the error you'll need to edit `apache2.conf`. To do so at the CLI type:

```
sudo vi
/etc/apache2/apache2.conf
```

And at the end of the file, add the line: "Servername <nameofyourserver>"

When you restart the Apache server you should no longer see the message.

And that is your basic LAMP - done and installed.

Unless you're a command-

line uber geek (there are a substantial amount out there) you'll prefer a more user-friendly method of LAMP admin. So, here's a run down of three favourite tools of the trade. As with the base stack, these installation advisories are based on Ubuntu.

## PhpMyAdmin

This is probably the most widely used MySQL DB Admin application. It's powerful and works very well. Install it as follows: At the CLI type:

```
sudo apt-get install
phpmyadmin
```

Accept the default bind message mid-installation. Open a browser and type:

```
http://<your server
IP>/phpmyadmin
```

input 'root' + the MySQL password created at install.

**TIP:** For external MySQL access using other management packages (such as MySQL Administrator), you must

remember that, by default, there's no access from any IP other than localhost 127.0.0.1. Unless you previously bound the database to a specific IP address, or opened it up to all by commenting out (#) the 'bind' line, you'll experience a 1130 error. To rectify this, just open up relevant user access in phpMyAdmin for the required IP number(s) - this stage is easy to forget and can throw you if overlooked.

## Webmin

Webmin is a quality utility, and the industry standard tool for fully comprehensive Linux server admin. The install runs as follows from the CLI (starting with the perl libraries):

```
sudo apt-get install perl  
libnet-ssleay-perl openssl  
libauthen-pam-perl libpam-  
runtime libio-pty-perl  
libmd5-perl
```

Next download the latest Webmin using the following command (at time of writing 1.480 is the latest release)..

```
wget  
http://prdownloads.sourceforge.net/webadmin/webmin_1.480_all.deb
```

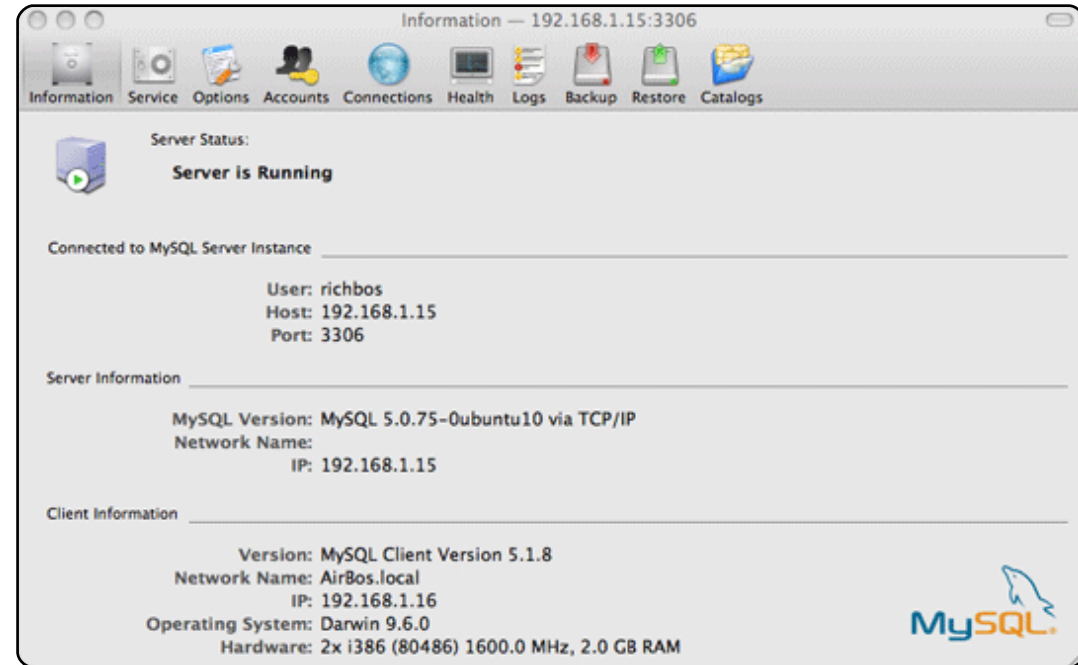
Decompress and install it..

```
sudo dpkg -i  
webmin_1.480_all.deb
```

**TIP:** Ubuntu doesn't allow logins by the root user by default. However, the user created at system installation can use "sudo" to switch to root. Webmin is quite sociable and will allow this user to log in. The Webmin URL is - <https://your-server-ip:10000/> Accept the certificate notifications and you should be in...

## MySQL Administrator

If you're not a phpMyAdmin kind of Guy (or Gal), you may like to try the somewhat more rounded and amiable GUI of the MySQL Administrator (see the image above right). This is a superb cross-platform tool, which you can download for free from here: (<http://dev.mysql.com/downloads/gui-tools/5.0.html>).



For more hardcore development, there's also the MySQL Workbench (again a free download). Get the Workbench here (<http://dev.mysql.com/downloads/workbench/5.1.html>).

OK, now we can administer our server. We're getting closer. In the next issue we'll discuss FTP and Firewalls.



**Richard Bosomworth** is a passionate Ubuntu professional who offers open source strategies and services for business IT from the <http://toomuchgreen.eu> web portal. When not working with technology he cycles vigorously.





# HOW-TO

Written by Adam Hunt

## Networking Ubuntu PC's With SSHFS

### SEE ALSO:

N/A

### APPLICABLE TO:

ubuntu kubuntu xubuntu

### CATEGORIES:



### DEVICES:



There are a number of ways to network two PCs that are both running Ubuntu, so that files can be transferred between them. We used to use USB devices and swap them to synchronize files between our two PCs, in what was euphemistically called a 'sneaker network'. One better way is to use an SSHFS or "Secure SHell File System" network.

SSHFS uses OpenSSH to provide secure (encrypted) communications between PCs. With the network connected, the local PC user will be able to transfer, open, and edit files on the remote PC, as if they were on her own PC. The remote files will look and act just like the files on the local PC.

To create an SSHFS network, all you need are two PCs connected to the same router or gateway. Ubuntu comes with most of the software to make this work, but needs two applications, available in the repositories, to be installed on both PCs. Here is how to set it up:

1. From System > Administration > Synaptic Package Manager install the following packages: **sshfs** and **openssh-server**.

2. Also in Synaptic, confirm that **openssh-client** is installed (usually comes with Ubuntu).

3. Go to System > Administration > Users and Groups > unlock > Manage Groups, confirm on *Fuse* group.

4. Create an empty folder in your home directory (Ctrl+Shift+N), it is usually best to name it after the other user, to prevent confusion.

That completes the installation of the networking tools. To activate the network from one PC to the other, you just have to invoke SSHFS and indicate where the files are coming from, and where they are going to be displayed (that new folder you made):

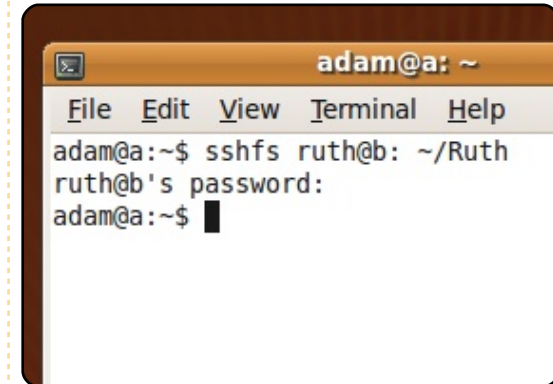
Go to Applications > Accessories > Terminal and enter:

```
sshfs
remoteusername@remotepc: /home
/remoteusername ~/newfolder
```

or, because SSHFS uses the other user's home directory as the default destination, you

can shorten the command to:

```
sshfs
remoteusername@remotepc:
~/newfolder
```



Once you supply the password for the remote PC (their password, not yours), which indicates that you have permission to access it, and, for the first time only, confirm the other PC, then the network will be established and an icon will be displayed on your desktop. You will be able to access the other user's home directory by simply clicking on that 'newfolder' icon in your home directory. You can then copy, move, edit, and delete files as if they were on your PC. Password-protected documents



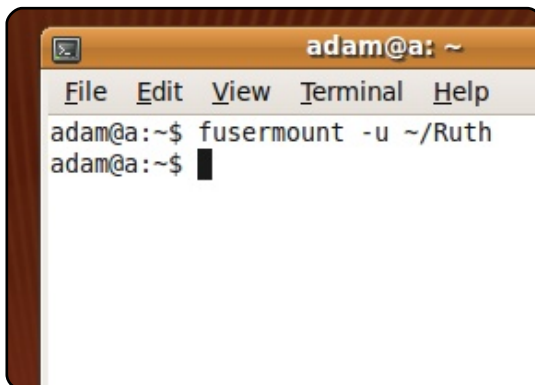
will, of course, remain password protected.

An SSHFS network is one way, meaning that each PC has to be individually connected to the other. If you connect to the other PC they won't be able to see your files without establishing a reciprocal connection to your PC.

To disconnect your PC from the network, enter in the terminal:

```
fusermount -u ~/newfolder
```

The '-u' means 'unmount'.



```
adam@a: ~  
File Edit View Terminal Help  
adam@a:~$ fusermount -u ~/Ruth  
adam@a:~$
```

There are a few pitfalls to keep in mind with an SSHFS network:

1. If either PC is rebooted,

then the network connection will be severed, and will have to be re-established.

2. If the network crashes through an error, then either or both PCs may lose their "Places" access, or other functions, until rebooted.

3. Another way to crash the network is by performing a ClamAV scan of the home folder while the network folder is active. Clam will try to scan all contents of your home folder including the contents of the remote PC, and that will create an error. The solution is to unmount (disconnect) the network first. This prevents the crash, and it results in a proper scan.

4. You can also crash the network by clicking on the other user's home folder when they are connected to your PC, and then clicking on their folder for your home directory and then back to theirs, etc. Don't do this!

There are probably other ways of crashing the network, which I haven't discovered yet.

There are a few pitfalls to keep in mind with an SSHFS network ...

Perhaps you will discover them. If the network crashes, just re-establish it from the command line or, if that doesn't work, reboot and try again. If changes are made to the other PC, such as its identity, or if you get a consistent error, then you may have to go to the hidden file in your home directory at `~/.SSH/known_hosts`, delete the data in that file, and then save it. Then, when you re-establish the network, you will recreate the data.

Try not to have both users, local and remote, edit the same document at the same time. If you have to work on a remote document, and you aren't sure whether the other user will be working on it, then it is better to copy it to your home directory and, after you have finished with it, copy it

back over to the other user's PC, overwriting the version there. You can check the remote version date and time to make sure that it hasn't been changed.

While it is a simple way to set up a network, SSHFS does work well and results in less wear on your sneakers than using USB devices to synchronize files between PCs.

I would like to acknowledge the help of the *Ottawa Canada Linux Users Group* for teaching me about SSHFS networking.



**Adam Hunt** had his first brush with UNIX in 1978, and has been using Ubuntu since April 2007. He lives with his wife, Ruth, in Canada, in a house with no windows. Their website is <http://web.ncf.ca/adamandruth/>





# HOW-TO




Written by Roger Wheatley

## Faster Internet With Squid

### SEE ALSO:

N/A

### APPLICABLE TO:

 ubuntu  kubuntu  xubuntu

### CATEGORIES:

 Games  Graphics  Internet  M/media  System

### DEVICES:

 CD/DVD  HDD  USB Drive  Laptop  Wireless

I've posted more than once about tips to improve our web surfing. This morning, I was cleaning out some old hardware (to take to the recycling centre), when I happened upon one of the last old laptops I had.

It's an older IBM Thinkpad T22, Type 2647, with 256MB of RAM and a 20GB hard drive. A couple of years ago, I had 20 of these units, bought from a recycling depot. I sold them all

on eBay, and have only a couple left now.



Whilst packing up the stuff to cart off, it occurred to me that I could put this old laptop to work by installing a proxy/caching server on it, and have my browsers pull much of the regularly requested web content off a locally cached network server. This means installing Squid. According to Squid's site, "...Squid is a proxy server and web cache daemon. It has a wide variety of uses, from speeding up a web server by caching repeated requests, to caching web, DNS and other computer network lookups for

a group of people sharing network resources, to aiding security by filtering traffic. Although primarily used for HTTP and FTP, Squid includes limited support for several other protocols including TLS, SSL, Internet Gopher, and HTTPS. The development version of Squid (3.1) includes IPv6 and ICAP support..."

I used the Ubuntu operating system for this project. Installing a Squid server on the network provided me with a few important benefits:

- Less bandwidth usage.
- Faster web surfing.
- Network cached copies of pages I regularly visit (if the original server is down).

Firstly, make sure you've installed a copy of Ubuntu 8.04 Server (Hardy) on the old laptop. Not sure how to do that? Here's a guide: *The Perfect Server - Ubuntu Hardy Heron* (<http://howtoforge.com/perfect-server-ubuntu8.04-lts>). In my

case I skipped (did not install) Apache, MySQL, Postfix, BIND9, Proftpd, POP3/IMAP and Webalizer. I didn't need them. I needed only the Squid proxy/caching server. I also installed Webmin (see below), so that I could easily manage this server remotely. A word to the wise: I learned (2 years ago) to install Squid first! This way the Webmin installation goes much smoother (I was using Debian for the server at that time and Ubuntu in another instance). Also, for those of you who have been following my blog (and for my welcome new readers), I also played with Squid and Ubuntu about a year ago, as described in the post "*Speed Up and Improve Web Surfing With an Ubuntu Squid Server*" (<http://ubuntulinuxhelp.com/speed-up-and-improve-web-surfing-with-an-ubuntu-squid-server/>). My earlier Ubuntu/Squid post was based on Ubuntu 6.06LTS and Squid 2.6. But things have changed, and applications, etc. have improved, so I thought a





# FASTER INTERNET WITH SQUID

revisit and reinstallation of the Squid server was in order.

I downloaded and burned a copy of the Ubuntu 8.04 LTS Server from Ubuntu's official site at: <http://www.ubuntu.com/getubuntu/download-server>. The bare minimum requirements are:

*300 MHz x86 processor  
64 MB of system memory (RAM)  
At least 4 GB of disk space (for full installation and swap space)  
VGA graphics card capable of 640x480 resolution  
CD-ROM drive or network card*

256MB of RAM made the installation slower than I'm used to. You can find more information on requirements for the Ubuntu Server (Hardy) at Ubuntu System Requirements (<https://help.ubuntu.com/community/Installation/SystemRequirements>).

After downloading and burning a copy of the ubuntu-8.04.2-server-i386 CD, I completed a base install of the Ubuntu server using the howtoforge.com guide above

as a reference. I also installed an SSH server so that I could tuck the old laptop away and complete everything else in comfort using my desktop.

```
sudo aptitude install ssh  
openssh-server
```

will get the ssh server up and running for you.

Throughout this post, you'll need to substitute your IP addresses and names to match those in your own network. After the installation of the base server is complete, open a terminal from your (comfortable) desktop and enter:

```
ssh root@192.168.1.200
```

192.168.1.200 is the address of the server I just installed.

Use the command

```
su
```

to enter root. That way you don't have to keep typing "sudo".

Install Squid using the command:

```
aptitude install squid3
```

After Squid has finished installing and you've rebooted the system, you may want to install Webmin, a GUI interface to manage that server. You can download a copy of the webmin package into any directory you like. The command to download is:

```
wget  
http://prdownloads.sourceforge.net/webadmin/webmin-1.470.tar.gz
```

Untar it like this:

```
tar xzvf webmin-1.470.tar.gz
```

Webmin needs Perl to run, so I installed some packages:

```
aptitude install install  
libauthen-pam-perl libnet-  
ssleay-perl libpam-runtime  
openssl perl perl-modules
```

Enter the extracted contents (of the webmin-1.470.tar.gz package) by typing:

```
cd webmin-1.470
```

Run the installation for webmin:

```
./setup.sh
```

I changed the port number from 10000 to 26395. I changed the admin account to "root", and entered a new password.

At the end of the webmin installation, I got the success message that included the URL: <http://squid.localdomain:26395/>

On my local PC, I had to edit my hosts file like this:

```
sudo gedit /etc/hosts
```

Then I added the following line:

```
192.168.1.200  
squid.localdomain squid
```

Now we want to reboot the Squid server using:

```
shutdown -r now
```

After rebooting the server, and logging back in (via ssh), you can see if the webmin service is running by using the

command:

```
sudo /etc/init.d/webmin
status
```

You should see something like:

```
webmin (pid 4573) is running
```

To see if it is listening on the correct port number, the command to check that is:

```
sudo netstat -tap
```

You should find a line in the output of the above command that says something like:

```
tcp 0 0 *:26395 :*: LISTEN
4573/perl
```

Remember, port 26395 was the one we chose to run webmin on - and webmin uses Perl.

Now open a web browser and visit webmin. The URL I would use is:

<http://squid.localdomain:26395/>

Again, remember that I added the appropriate information to my hosts file so

that the browser can find the URL.

I chose not to enable SSL for logging into webmin, as I don't need it in this LAN. After logging in, we want to configure Squid. Look for something (on the left) that says "Unused Modules" and look for "Squid Proxy Server" - click that link. You will see an option to install the Squid (webmin) module. Select that link to install it.

After installing, look on the left side menu, and, under "Servers", you will see "Squid Proxy Server". Select "Squid Proxy Server" and then select the "Ports and Networking" option.

Note that Squid is running on the default port 3128. Now return back to the Squid module page by clicking "Module Index" (at the top of the page). Select the "Access Control" icon and see a button at the bottom of the page that says "Browser Regexp" - that contains a drop-down list. Use this list to select "Client Address," then click the button

that says "Create new ACL".

Enter your values in the form. I used the following:

```
ACL Name: localdomain
From IP: 192.168.1.0
To IP: 192.168.1.255
Netmask: 255.255.255.0
```

I didn't change anything else, and clicked "Save".

Now click the tab (at the top) that says "Proxy restrictions". Click (at the bottom) "Add proxy restrictions" and look for the new ACL name you just created (mine was called "localdomain"). \*\*\* Make sure you are looking under the column that says "Match ACLS" \*\*\* and click on this name. Now click the radio button that says "Allow". Then select "save" at the bottom.

In the new screen that displays, use the up arrow to move "localdomain" (or whatever you called your new ACL rule) so that it is just above the line that says "Deny all". If you don't, your browsers will not be able to get access.

Now in your ssh window, use the command:

```
shutdown -r now
```

This will restart the server and Squid 3 (along with the new configuration). I noticed that webmin (for some strange reason) thought Squid 2.6 was installed. Therefore, webmin was unable to start the server. But after rebooting the system, the "Stop Squid" button appeared - so I assume the webmin module has started working properly (no need for me to play with webmin again, as I'll use SSH to access and reboot, etc., so I did not try).

As a final step, make sure that you set the proxy server address in your web browser. In my case the information to enter as a proxy server for each web browser is:

```
192.168.1.200:3128
```

In July 03 I found a problem when trying to access statistics. Here's the issue and fix:

## Issue:

When trying to access the "Cache Manager Statistics" under "Squid Proxy Server", the following error displays:

**"The Squid cache manager program /usr/lib/cgi-bin/cachemgr.cgi was not found on your system. Maybe your module configuration is incorrect."**

## Fix:

```
aptitude install squid-cgi
```

Cache manager statistics will now work.

## Other sites with related information:

*How to Block websites using Squid Proxy in Ubuntu Linux*

<http://shibuvarkala.blogspot.com/2008/11/howto-block-websites-using-squid-proxy.html>

*How to Block a Port in Squid Proxy, Ubuntu Linux*

<http://shibuvarkala.blogspot.com/2008/11/howto-block-port-in-squid-proxy-ubuntu.html>

*Installing an HTTP proxy server (Squid)*

<http://en.kioskea.net/faq/sujet-804-ubuntu-installing-an-http-proxy-server-squid>

*Installing Squid Proxy using Webmin on Ubuntu Server 8.04.1*

<http://chrisjohnston.org/2008/installing-squid-proxy-using-webmin-on-ubuntu-server-8041>

*Paranoid Penguin - Building a Secure Squid Web Proxy, Part I*

<http://www.linuxjournal.com/article/10407>

Enjoy faster web surfing as less external files are requested for pages you commonly visit, and local cached copies are delivered to your browser. I hope you ladies and gents have fun playing with this, and I hope it helps you out.

## A QUICK OPINION

When I buy a new product, I often feel the need to justify the purchase to myself and others. Why would I want to believe that my new purchase could have been a waste? It is human nature to convince myself that I made a good decision.

My friends who use Windows or Mac OSX feel the same about their purchases. They purchased the computer, maybe even paid for the operating system separately, and want to get the most out of their investment. There is no reason to expect them to be quick to go back on their decision and toss their investment out the window.

Those of us who use Linux desire that same sense of validation. Many of us made that leap of faith and left behind an expensive proprietary operating system. We have much to gain in convincing our peers to do the same, but sometimes we are our own worst enemies. When what those peers need to hear from us are valid reasons to make the switch, all they hear are cries of "Jump! Jump!" The solution is simple: give them what they want — make them jealous.

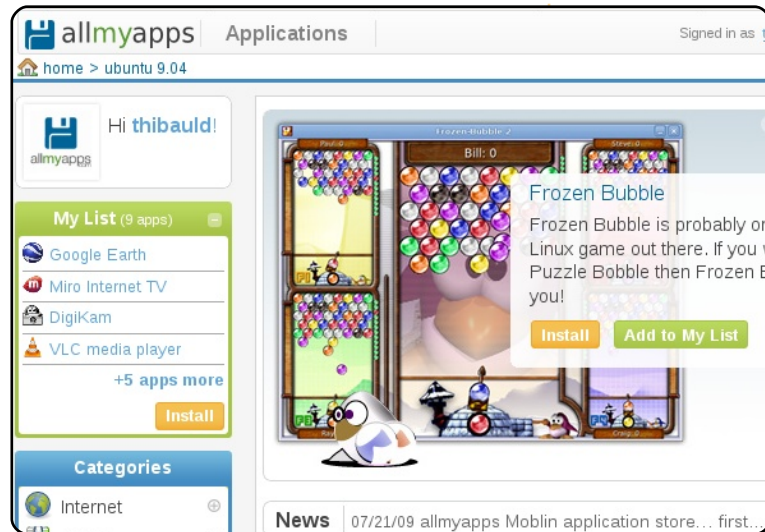
It is not a secret that Windows is prone to crashes. For someone whose browser crashes while trying to check their email, the ability to complete such a routine task without having to worry about crashes is a dream. Someone else, stressing over how to pay for a new install disk to repair their damaged operating system, would be relieved to know that they can legally install a new operating system for free, repair it for free, and distribute it for free. Someone tired of combating viruses would drool at the thought of running a computer safely without the need for a virus scanner, and a person hesitant to pluck down money for Microsoft Office would find a lot to appreciate in OpenOffice.org. We do not need to preach these benefits to others.

At the same time, some people just want to be wowed. When they see a window wobbled, a user shift through a ring of open windows, or a desktop flip to reveal another workspace, their appetites get whetted. All these features can be performed on Windows, but in most cases they will cost money, and their performance will still fall short of that offered by Compiz.

Keep the grass on your side of the fence as green as you can, and people will eventually jump over.

**Bertel King Jr.**





**Allmyapps** is a Web-based application, discovery, and installation tool for Ubuntu that I've been working on, with a friend, for the past nine months.

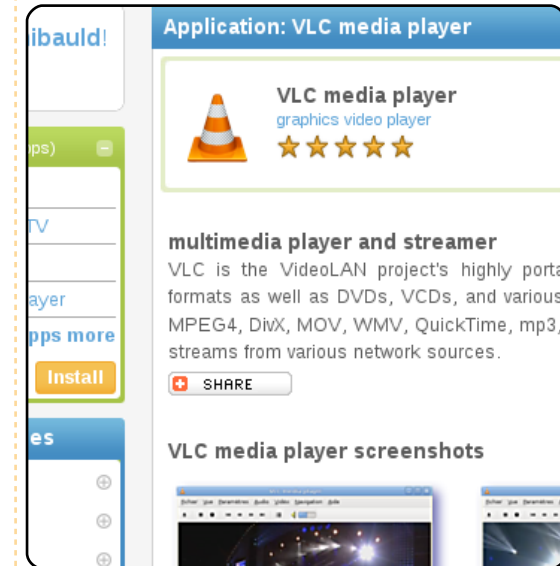
I get frustrated each time an iPhone owner talks about how cool the Apple AppStore is. Technically, the Apple AppStore is nothing but a package repository like those we've enjoyed for ages in the Linux world! Our goal is to unleash the true power of Linux package repositories, and show it to the world.

the depth of our package repositories to provide users with a tool that is fun, easy, and, of course, secure! We have tried to make allmyapps visually appealing, so that people will enjoy browsing applications on it. In the same spirit, we have also focused a lot on usability to help people find the applications they need, either by browsing categories or by performing a free search. Finally, as allmyapps relies on the apt package management system (through apt-url), it is completely safe for users to use.

With allmyapps, we want to give users the best application installation tool in terms of eye-candy and usability. Indeed, allmyapps leverages on the power and

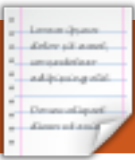
A lot of people asked us why we decided to implement allmyapps as a Web application and not as a traditional desktop application. The main reason is that we want allmyapps to be visible to all users (not only Linux users). This is also why, beyond traditional social features like screenshots, reviews, and ratings, allmyapps gives you the ability to create lists of applications that you can share with others via email or a Web widget. The goal here is to help spread the word and make people realize how easy and fun it is to install applications on Linux!

In the coming months, we hope to expand allmyapps support to other systems. We chose to focus on Ubuntu first because it was the most desktop-oriented Linux system in our opinion, but, eventually, we'd like everybody to benefit from allmyapps.



In conclusion, if you like our project, we invite you to visit allmyapps to create and share your list of favorite applications! And if you are in the mood for contributing, do not hesitate to rate, review, and upload screenshots of your favorite applications. Of course, any feedback is warmly welcome. Feel free to contact me at [thibault@allmyapps.com](mailto:thibault@allmyapps.com).  
**Thanks!**

**Source:** <http://allmyapps.com>



# MY STORY

Written by Stefano Colecchia

## My Linux Experience - Story 1

**A**t the beginning of 2005, I subscribed to ADSL with an Internet provider. At that time, I was using Windows on my PC, and I spent my winter holidays reformatting and reinstalling the OS, I think at least a couple of times, because every time I accessed the Internet I got a virus.

I was frustrated by this situation and did not know how to solve it. A possible solution was to install anti-virus software, but I did not want to spend the money for it. At that time I did not know that ClamWin anti-virus existed - actually I did not know about the existence of any open-source software. In addition, I did not want to slow down my four-year-old PC with anti-virus software.

Eventually, I cancelled the ADSL contract and reported my experience to one of my colleagues. He was a Linux fan, and he mentioned a new

distribution named Ubuntu. He had just received a bunch of CDs from the company that distributed this OS. So, he gave me a couple of these. The distribution came with 2 CDs: one live CD and one install CD. My colleague suggested that I try the live one.

I inserted the live CD of Warty Warthog. My first experience with the live CD was exciting, and I just played with it for about a week. Soon I realized I wanted to broaden my Linux/Ubuntu experience and decided to install Ubuntu. Since I also wanted to keep the Windows OS, I opted for a dual-boot system. Due to my inexperience, I managed to install Ubuntu, but I failed to create a dual-boot system. Windows was still present in a partition, but whenever I selected it from the GRUB menu, it failed to start up.

I was both happy and desperate. Happy because I had Ubuntu installed,

desperate because I was not able to perform my routine work without Windows. So, after a couple of weeks, I decided to reinstall Windows. Anyway, I did not want to surrender, so I visited the Ubuntu international forum (at that time an Italian Ubuntu forum did not exist) looking for advice on how to obtain a proper dual-boot system. Luckily I found various threads that explained how to perform this operation. I was encouraged, and decided to try the install process again. This time everything went well and I was so happy! I could choose to start Windows, or to use my new wonderful Warty Warthog.

Obviously, I started spending more and more time on my Ubuntu box rather than on Windows. For me, Linux was a new world ready to be uncovered. Everything about this OS was fascinating: the Gnome desktop, the applications, and even using commands from the Terminal!

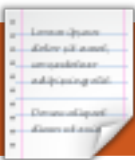
Step by step I started to perform my routine work with Ubuntu instead of using Windows. Obviously I spent hours surfing on the Ubuntu forums and Wikis to see how to solve my problems. I learned many things: how to play DVDs, how to install multimedia codecs, how to connect to the Internet, how to burn CDs/DVDs, etc.

I was able to overcome obstacles that, at a first glance, appeared insurmountable. I also started to give my own advice to people asking for help in the Ubuntu forums. And, what satisfaction when someone thanks you because your advice is successful!

Currently, I am multi-booting between Debian Etch, Hardy Heron, and Intrepid Ibex.

I think that, in the coming years, I will always use this beautiful Operating System, which is Linux.





# MY STORY

Written by Bill Chan

## My Linux Experience - Story 2

Ubuntu 8.04 really amounted to a quantum leap for me - a godsend. The operating system industry irked me so much that I endured 12 years of using Microsoft Windows 95, since I accepted that radical OS in 1997. I refused to do any Windows OS after Windows 95. Linux was on my mind for over a decade.

For me, Windows 95 pushed out the failed IBM OS/2 product I had learned to adore so much, which itself had previously pushed out DOS 6.22 + Win 3.1. A single Intel P233MMX-based ASUS VX-97 motherboard ended up covering me for almost 13 years, right up to its very recent retirement in January 2009. The attempts at improving Windows 95 via Windows 98, 98se, Me, and XP, were all disappointments to me. My workplace experience with Windows NT 4.0 and then Windows 2000 told me to steer

clear of those as a home OS. The fiasco of an embedded browser that started with Win98 was an immediate concern about what was to come in future years, and the progression of Windows to a big mess did in fact occur. Windows became a big, vulnerable OS, requiring 3rd-party software to maintain it and rid it of viruses and other malware. I resisted letting go of Windows 95 until an appropriate replacement OS arrived. It turned out to be Ubuntu 8.04 LTS.

It was a serious sacrifice for 12 years to endure lack of USB support, and the inability of Windows 95 to support new hardware and software - but I was always able to get by, just barely. I would have had to drop Windows 95 had it not been for the Opera browser. Firefox did not support Windows 95, but Opera did. However, overall, the system was crippled due to the seriously dated Windows 95.

Adobe Acrobat 5.1 was the last version for Windows 95, and an old Y2K edition of ZoneAlarm firewall was all that was available to help protect my Internet connection - via dial-up. It was pointless to do broadband with Windows 95, given that Shockwave and Flash no longer offered upgrades to that seriously dated OS. Without proper Flash support, it hampered playback of streaming video. The slow hardware, with a PCI video card, would have seriously faltered - even with proper software support. The OS had to be upgraded, along with modern hardware, to make sense of going to broadband. Despite all this, I waited for the proper distribution of Linux to come along and give me the path to revolutionize my often laughable system. Thanks to a very busy work schedule, there was little time to really use my PC. But once I retired, that old platform and OS quickly became unbearable.

The serious difficulty I had was that my old 128-MB computer was so old that it couldn't even support Linux, so trying out Linux over the years would have required fussing with another computer, which I did not want to do.

And it would just be an interim solution; the Linux distros were not what I wanted. I had access to Linux installations from friends, and read about the developing Linux distros on the Web for many years. When I read that even early editions of Ubuntu had trouble dealing with wireless LANs, and even playback of MP3s, I concluded that Ubuntu was not ready for prime time. I did not want to do a part-way upgrade, but rather a full upgrade. Mandriva Linux was considered the most likely target distro at the time - as Windows 95 got so unbearable. In the end, I opted to endure Windows 95 and waited for a Linux distro I really could be happy with.



At no point was MacOS or Windows XP considered. Microsoft is about monopoly, and Apple would be even a more ruthless monopoly if it could get there. I believe in open-source software, Internet access for all, and in paying for only the support of the infrastructure, not in paying monopolies focused on controlling home users and supporting their business model and their way of computing. I refused to even consider getting new hardware and Microsoft Vista. From desperation, via rumors, Windows 7 sounded viable, but that could only be a risky proposition and giving in to Microsoft.

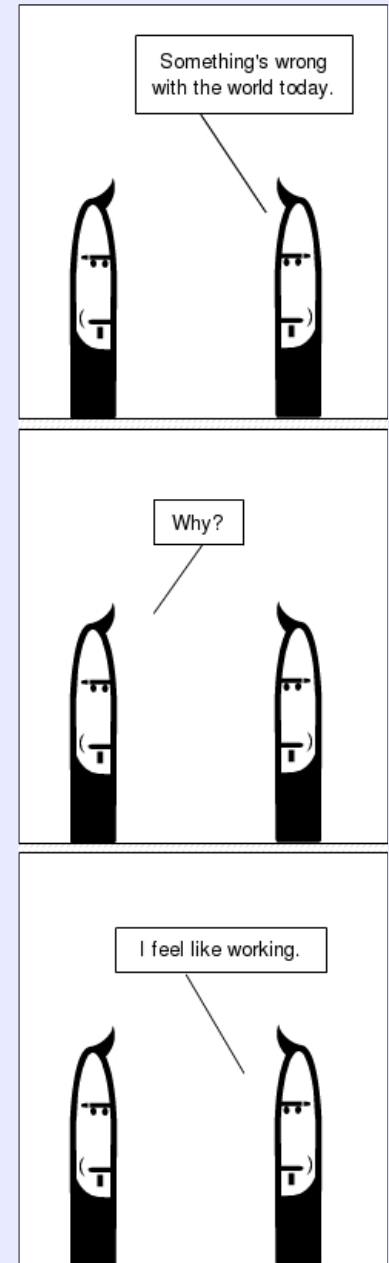
Just as I was going to try Mandriva Linux on a 512 MB Pentium III system, salvation came when reviews of Ubuntu 8.04 LTS were digested! As if by magic, Dell offered a modern netbook Mini 9 with Ubuntu 8.04 pre-installed. If Windows 95 sufficed for my needs, I concluded that a modern dual-core 1.6 GHz

netbook, with Ubuntu 8.04 LTS, would surely fit my needs. It did!

Open Office is almost completely compatible with my Office 97 Word and Excel, offering immediate bridging of key functionality. Video playback was complete in Ubuntu 8.04. Wireless LAN support was immaculate, to the point where the January 1, 2009 transition to the netbook never saw my Ubuntu-Mini 9 on dial-up Internet access. The quantum leap abruptly ended the trusty P233MMX and Windows 95 duo. Never has the gestation with a new OS been so complete and so quick.

Using WiFi and DSL, Ubuntu - in just a month - revolutionized my computing experience. Thanks to immense and complete information on the Internet, I became a power user in just a month, though both abundant time (thanks to retirement), and experience in workplace Unix, certainly helped! My Ubuntu 8.04 netbook is lean and mean, by means of

tweaking all aspects of logging, effective backup using tar, cleaning out foreign language support and help files, and more customizations and tweaking than I could ever write about. Clearly, Ubuntu 8.04, with the wealth of information on the Internet, offers a solid viable approach to computing, which permits users to avoid the trap of computing the expensive and guarded Apple way or the controlling and painful Microsoft way. Kudos to Ubuntu 8.04 and Dell support.



*Richard Redei*

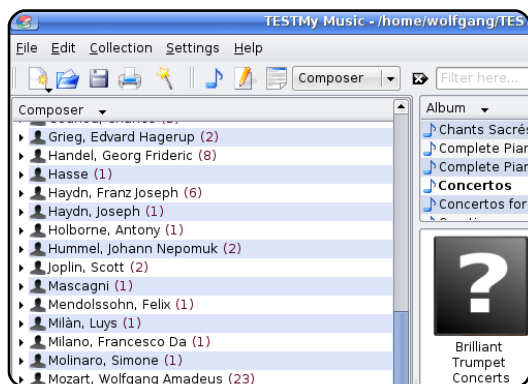




# REVIEW

Written by Melanie Fogel

# Tellico



There are very few people who don't have a collection of something: books, music, movies, matchbook covers, rocks, girlfriends. At some point, that collection will grow to where you can no longer keep everything sorted in your head. That's where Tellico comes in. It's available in the Ubuntu repositories since Dapper, and the homepage is at: <http://tellico-project.org/>

What makes this review different, I hope, is that I hold a degree in library science, and was involved in the development of one of the first -- if not the first -- library

cataloging applications for desktop computers. I'll be talking about music, but Tellico can catalog anything from stamp collections to comic books. And it's lightweight -- only about 9MB fully installed.

I decided it was time to catalog my classical-music collection. Classical music is always a nightmare to catalog, even for public or university libraries, because, for example, the cataloger must deal with multiple recordings of one piece of music -- even by the same performer(s) -- and a recording of any piece of music can be bundled with pieces different from those in other CDs. In addition, conductors may record the same symphony with various orchestras, and there may be piano versions of symphonies and symphonies with soloists, and so on. So, classical-music lovers with large collections can use help in determining whether a piece of music, or a particular performance of that piece, is in their collection

before they buy some CD that has caught their eye (or ear).

The most difficult thing about databasing or cataloging (I'll use the terms interchangeably) is setting up the structure. In the bad old days, once you had established the information categories ("fields") in your database, you couldn't change your mind without setting up a new database and then re-entering the data from scratch. With Tellico, you can add, subtract, or reformat fields at any time, although you'll probably have to manipulate your data, since no database program can guess how to shift things about. I recommend that you start with 50 to 100 of the items you want to catalog. By the time you've entered all that data, you'll have a pretty clear idea of the possibilities of what kind of information you might want to access in the future.

Tellico comes with several sample collections. You might want to start with one of them -

or at least explore how the fields are set up - before you embark on a database of your own. Tellico supports eleven types of fields. These are well documented in the excellent help files. The only one I had a little difficulty understanding is the Dependent field. Clicking around in the sample collections, I discovered how to use it in the wine collection. Basically, if you have several items with, say, identical titles, you might want to distinguish them by year. Creating a Dependent field that contains references to the title field and the year field will produce a list of titles with their appropriate year. You'd want to keep the title and year as separate fields so that you can join them to other fields, or list them by themselves.

For those of us with insistent friends, Tellico allows your loans to be tracked by recipients and dates, and it also allows the insertion of reminders (using KOrganizer) to ask for payment. I'll be



making heavy use of this once I get my personal library into Tellico.

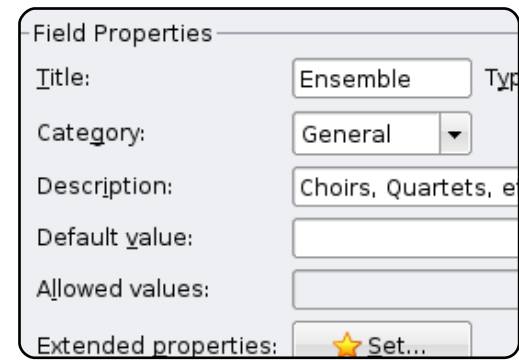
If you so choose, most fields can contain multiple entries. For example, if there are several performers on the same CD, you can enter them all in one field, separated by semicolons. If, however, you want those entries to be listed separately in an alphabetical sort, you have to configure that field to format as a name or title, even if it isn't. There's no real negative consequence to this, unless you're anal about capitalization. And you can create exceptions to capitalization in the basic Tellico settings -- just make sure you don't use terms that should be capitalized under other circumstances.

Tellico can import data from Amazon, IMDb, and Cddb, as well as from other file formats. It can also export to various file formats. I haven't tried any of these features yet, except for importing this database into Tellico from OpenOffice spreadsheet after several failures at getting the database

set up (in the latter) the way I wanted.

One disappointment I have with Tellico is the Table field. Like all of Tellico, it's highly configurable. However, it's not easily editable. That is, if you enter data and make a typo, you can't correct the one letter. Clicking in any of the table's boxes erases all the data in that box. Nor can you paste text copied from somewhere else. I searched the Web and found no one complaining about this shortcoming. If it's a bug in the current version, it's as yet unreported.

Another disappointment is Tellico's limited ability to generate reports. Many of the fields in my music collection have multiple entries -- for performer or featured instrument, for example. While I can, with a click of the mouse, generate a list of names or instruments, I could not manage to generate a printable report of this information. Tellico's reports contain the fields one chooses to display at the top right of the screen. When there are



multiple entries in that field, they display together unsorted, and that's how the reports come out. For most people this is probably not a huge problem, but the works of classical composers often have catalog numbers so that one can tell one concerto grosso from another. Having a printed list of album titles sorted by these numbers - that I could keep in my wallet - could save me some money when I'm in the CD store, because I'd know from the list if I already own a copy of the work. It may be, however, that I just haven't explored all of Tellico's possibilities yet, or familiarized myself with the program sufficiently to invent a workaround.

If you've never used a program like Tellico before,

prepare for some frustration -- not with the program, but with yourself for not predicting everything you're going to want in the database, or how you'll want to access it. That's quite normal, even for people like me who've been working with other people's collections for decades. Tellico's preconfigured sample collections will cut down on the frustration considerably, although that's hard to appreciate unless you've first tried to set up a database from scratch.

On a final note, Tellico would be great for indexing your copies of *Full Circle*. All you'd have to do is set up fields for Issue Number, Author, Title, Subject, Page Number(s), a yes/no box for whether the article is illustrated, and whatever else you might search for in back copies of *Full Circle*. Tellico comes with the ability to add ratings for the quality of the item you're cataloging/indexing, so you could even rate each article to know at a glance whether it's worth re-reading. Oh my! The possibilities are endless!



# MOTU INTERVIEW

Taken from [behindmotu.wordpress.com](http://behindmotu.wordpress.com)

## Stephane Graber

**Behind MOTU is a site featuring interviews with those known as 'Masters of the Universe' (MOTU). They are the volunteer army of package maintainers who look after the Universe and Multiverse software repositories.**



**Age:** 18

**Location:**  
Sherbrooke, QC,  
CA

**IRC Nick:**  
stgraber

**How long have you used Linux, and what was your first distro?**

I've been using it for almost 9 years now. My first distro was Mandrake Linux; I then switched between distros for a few years including Gentoo, Corel Linux, Caldera, LFS, Slackware, Red Hat and Debian - to finally use Debian, and then switch to Ubuntu.

**How long have you been using Ubuntu?**

Since Warty.

**When did you get involved with the MOTU team, and how?**

My first package to be uploaded to the archive was pastebinit - a command line pastebin client; this one was uploaded on the 7th of December 2006 (feisty).

**What helped you learn packaging and how Ubuntu teams work?**

I mainly learned how Ubuntu works through discussions with people on IRC, and reading some wiki pages. Having been around for a long time, I actually saw most of these team structures being created.

For the packaging, I initially simply read the Ubuntu Packaging Guide, then followed the REVU process doing changes when needed.

**What's your favorite part of working with the MOTU?**

Being able to get my favorite

applications directly in Ubuntu, fixing some others, and helping others to do the same. And, of course, long discussions on IRC :)

**Any advice for people wanting to help out MOTU?**

Just go on IRC, ask people, and read some wiki pages. Packaging is not as hard as it may look.

**Are you involved with any local Linux/Ubuntu groups?**

I recently moved to Canada where I'm not involved in any LoCo, but I'm still one of the admin of the Ubuntu Switzerland LoCo team. My role is now to mainly follow the discussions on the mailing-list and on IRC.

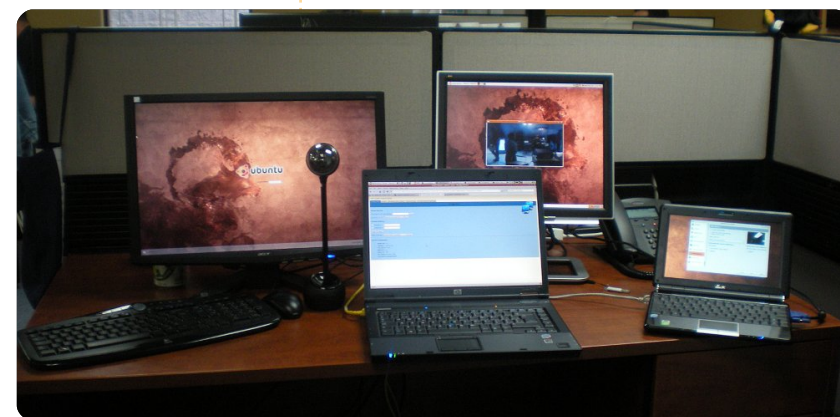
**What are you going to**

**focus on in Jaunty?**

Improving LTSP to get desktop-like experience, getting ltsp-cluster uploaded to Universe, improve a bit iTalc's integration with LTSP, and getting more educational packages uploaded/improved for Edubuntu.

**What do you do in your other spare time?**

I work on upstream LTSP, and on some of my personal projects (pastebinit and a few others). Yeah, I don't really do anything beside working on open source projects.







# LETTERS

Every month we like to publish some of the emails we receive. If you would like to submit a letter for publication, compliment or complaint, please email it to: [letters@fullcirclemagazine.org](mailto:letters@fullcirclemagazine.org).

**PLEASE NOTE: some letters may be edited for space reasons.**

I read the Amarok 1.4 review by Damien McGuignan (FCM #27), and I think there should be some clarifications and corrections:

1. The article starts with: "Version Amarok 2.2 is now out". This is not correct, the latest stable release of Amarok is 2.1.1, available in the jaunty backports repository. But there are good chances that Amarok 2.2 or at least a beta version of it will eventually be shipped with Karmic.

2. He states this is a review of Amarok 1.4, which is most unlikely from the screen shots I can see, and it would be quite old and outdated. I think this should read "Amarok 1.4.9.1", a version released on April 12, 2008, whereas version 1.4.0 was released on May 17, 2006. There is nowhere in this article I can see anything stating that this will \*not\* be the 1.4.0 version.

3. He suggest a downgrade to version 1.4 from version 2.0.2 that was shipped with Jaunty, but omits several important details:

3.a. Amarok 1.4.x is not maintained anymore by its developers, and the maintenance by the Kubuntu packagers of Amarok 1.4.9.1 shipped with Hardy will end in October this year, the Kubuntu version of Hardy being not an LTS release. This will leave the users pretty much alone with the software.

3.b. Installing software from a PPA is installing unsupported 3rd party software, there is absolutely no support given from the Kubuntu Community, unless some volunteers step in.

3.c. There is no support offered by the packager of this PPA, which should be clearly stated in this article. It should also be clearly stated that neither the Amarok developers nor the Kubuntu Ninjas will

## LETTER OF THE MONTH

Writer of Letter of the Month wins two metal Ubuntu case badges!



I recently finished my elementary school and am getting ready for high school. The computer lab in which I learned was an awful place with old computers running Windows 2000. Recently, in one of them, Windows didn't work and the teacher asked me to format it. I told her all about Free Software, GNU and Linux and she said let's try it, so I installed Ubuntu on that PC.

The teacher started learning Ubuntu and Open Office and said it's very easy. I am now the leader of a Linux User Group. Some of the computers I installed Ubuntu on had 64MB RAM, 6GB hard drives, 8MB graphic cards, old IBM machines.

Some students asked the teacher how to get Ubuntu, and the teacher directed them to me, so they became part of our LUG (Linux User Group) here in Gjakova. We are holding weekly meetings. Some of these new users are amazed, not only by seeing Compiz, but most of them converted because of the fact that there are no viruses.

Some friends I converted earlier

are helping me converting others. The hardest thing to convert are the gamers. The other people (those who use MSN Messenger, a browser and Facebook) are easier to convert. So, now, that's how I spend my free days, helping people with Ubuntu and making them legal because in Kosova they don't respect the copyright laws.

I am also working for a free software conference that's being held in Kosova this year.

**Heroid Shehu**





accept any reports or complaints regarding this downgrade.

**Myriam** aka Mamarok

Ed: *While I agree with a couple of your points, I think that it is quite fair to shorten 1.4.9.1 to 1.4, many people (including yourself) use 1.4.x. Regarding the use of 1.4, I personally use 1.4, and much prefer it to the current Amarok. Yes, I'm using a version which is no longer supported, but that (to me) is the beauty of Linux: having such a wide choice in which applications I wish to use. Even if it is an older version.*

In issue 27 CW Moser says that the major pain of Virtualbox is in having to re-compile the kernel every time a new version comes out. There is a program called DKMS, available from Synaptic, that does the work for you. Ever since I discovered this, I have had no problems with new versions of Virtualbox.

**Chris Burmajster**

I've been looking for a video capture program that works under Ubuntu (running version 9.04) that is similar to FRAPS. If unfamiliar: FRAPS is the most prominent program to capture in-game action for MMO's or other computer games in Windows.

While FRAPS works under WINE for some programs, I am trying to find a native version that will work on both WINE, and non-WINE, games. While there are various screen capture programs that take snapshots, I am coming up blank on video capture programs.

**Brian Jenvey**

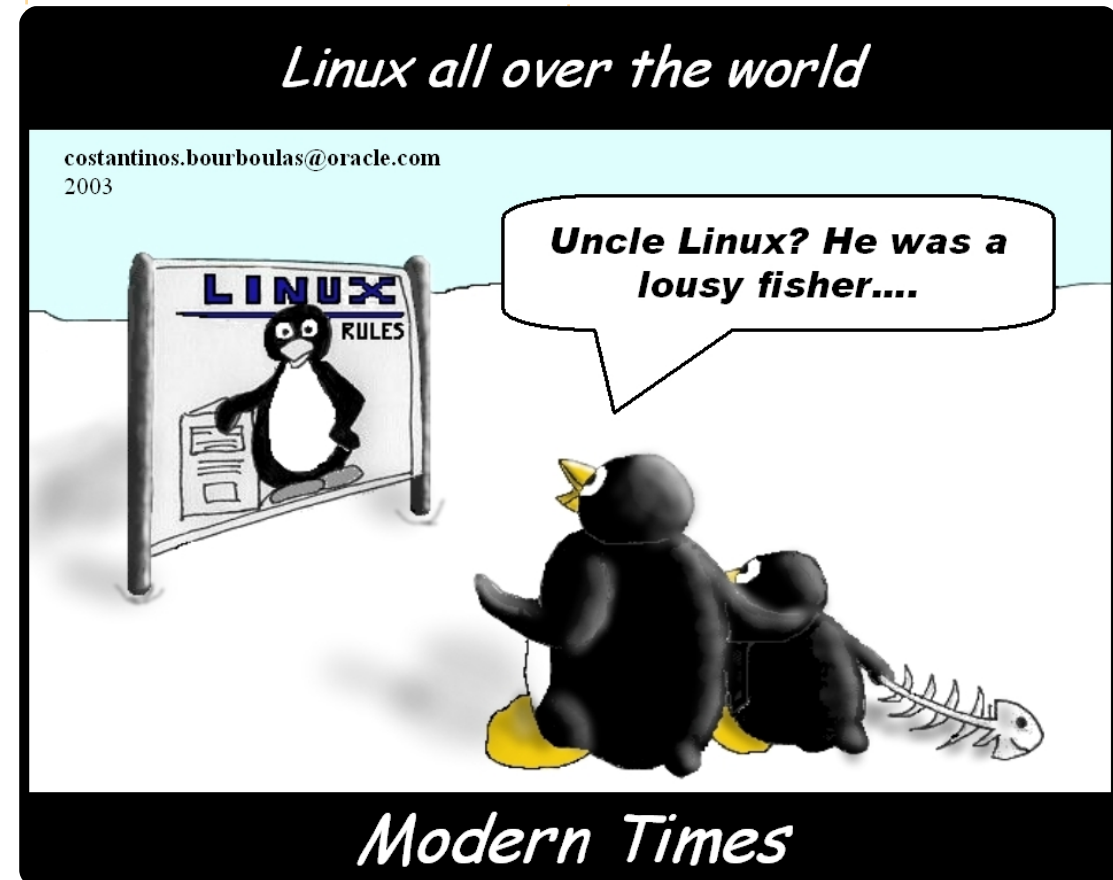
I've been using Ubuntu for the last two years and I have completely settled on Ubuntu and am very happy with the developments, stability and (most importantly) the user friendliness of the system, which was lacking in earlier flavors of Linux.

In the past, Linux had overcome most of the barriers it was having due to hardware support etc., but one important point which is significantly still lacking is a good quality, rapid application development database like Microsoft Access. Open Office Database is good, but still lacks the functions available in Access, such as Forms, SubForms and such-like.

It would be great if you could start a tutorial on desktop databases, as I think it would be very useful for all day to day Linux users.

**Anura Senarathna**

Ed: *Any database ninjas out there who could write us an article (or several) on databases?*





**Amber Graner:** Leigh, welcome, and thank you so much for participating in this month's "Women Behind Ubuntu" interview - Please tell us a little about you.

**Leigh Honeywell:** Hi Amber, thanks again for interviewing me! I live in Toronto, Ontario, Canada, but travel excessively. By day, I wrangle Windows viruses for a major anti-virus company - obligatory disclaimer here that the views below are mine, not theirs. I'm also a part-time student

finishing up my BSc in Computer Science (<http://web.cs.toronto.edu/>) and Equity Studies (<http://www.utoronto.ca/equitystudies/>) at the University of Toronto, and am one of the founders as well as a board member for HackLab.TO, Toronto's hackerspace. In my copious free time I'm an avid and eclectic reader, a cyclist, and have recently started running.

**AG: How did you get involved in F/OSS, and Ubuntu specifically?**

LH: In high school, I was involved in a group called the Ottawa Carleton Educational Space Simulation ([spacesim.org](http://spacesim.org)). We had a mock "space station" and ran various flavors of Linux - mostly Mandrake. I got my first taste of \*nix there, but it didn't really catch. I asked for a Mac when I went off to University to study physics.

In my third year of university, I

decided that I'd learned all I could about Mac OS, sold my Mac, and bought a bare-bones laptop with no OS preloaded. Four months, untold reinstalls, distro changes, kernel compiles, and even a few patches of my BIOS later, I had a working Ubuntu laptop (and had dropped out of school and started working for a telco). It was an amazing, immersive, and maybe a little obsessive way to start using and learning GNU/Linux, and I've been hooked ever since.

**AG: What is it that you find engaging about Ubuntu?**

LH: From a purely practical perspective, I am at my most productive on Ubuntu. I use Red Hat extensively at work, but my desktop and netbook run Ubuntu.

On the meta level, I love the community which I've found among Ubuntu users, from the Canadian LoCo team to Ubuntu Women to the joy of popping into #ubuntu on Freenode to

ask one question, and answering five others.

**AG: I noticed you ran a workshop at the University of Toronto during the <https://www.gr8-designs.ca/>. Are there other programs designed for girls/women that you are involved with in the Ubuntu community, and/or the F/OSS community as a whole, that you would like to mention and tell us about?**

LH: Gr8 Designs for Gr8 Girls was amazingly fun. I had the privilege of introducing nearly a hundred eager grade 8 students to the Arduino micro-controller platform. We talked about the idea of making the things around us smart by putting computers in them and then we made LED throwies, which the girls seemed to really like. Part of what makes Gr8 Designs such an important program is that we're talking to girls at a young age, where they have been exposed to



fewer stereotypes about who "can compute". Instead, they get to meet and learn from a bunch of women who are working in the field, and doing all sorts of neat things with computers.

I've been involved for a couple of years now with the Ubuntu Women project, which is a fantastic community of women and allies who are interested in helping solve bug #1 both by getting more women involved in Ubuntu, and creating a supportive environment for those of us who are already involved.

I've known about LinuxChix for years, but only recently signed up for the mailing lists and started lurking in the IRC channel, and have found a great deal of support and general awesomeness there as well.

Next year, I'd really like to attend the Grace Hopper Celebration of Women in Computing, which the Anita Borg Institute runs, but this year it's a little too close to SecTor (sector.ca), the security

conference I'm helping out with in Toronto.

**AG: Karmic Koala, Ubuntu 9.10 is due the end of October, what excites you about this upcoming release?**

LH: I'm looking forward to seeing the results of all the work that's been done on improving boot speeds. I've been pretty happy even running whole disk encryption on my netbook, but it'll be awesome when booting up takes less time than it does for me to enter my absurdly long pass-phrase. I'm lucky that my netbook (the MSI Wind) has had effectively perfect support since 9.04, but improved support there is going to even further lower the barrier to entry for Linux on portables.

I think the replacement of Pidgin with Empathy is going to be a huge and interesting change, but I really hope the Empathy folks devote some time to implementing Off-The-Record Messaging (cypherpunks.ca/otr) in the client soon. Until then, I'll be

sticking with Pidgin. I understand how, for the "mainstream" / less paranoid user, this isn't a vital feature, but I think it should be :) (I should state my bias in the above, though - I'm a mentor in the Google Summer of Code program this year, and my student is working on OTR.)

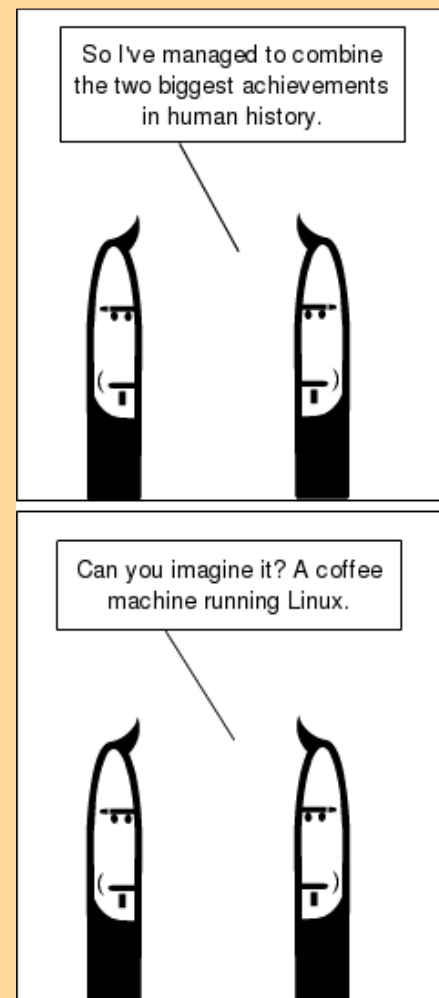
**AG: I am sure there are many things about GNU/Linux and Ubuntu that you would like to discuss and that you are passionate about. Is there anything you would like to discuss or inform us about?**

LH: I'm really passionate about my local community, and I'd love to see more Ubuntu folks get involved in hackerspaces and more hackerspace folks in Ubuntu. I think there are some great opportunities there to cross-pollinate volunteers, and gain different perspectives. hackerspaces.org is a great resource for finding (or starting!) your local hackerspace - there may be one closer than you think!

**AG: Leigh, Thank you so**

**much for taking the time giving us a glimpse into what it is about the F/OSS - and the Ubuntu community that you enjoy and work on.**

LH: Thank you, Amber!



Richard Redei





## GAME NEWS



### • id may stop supporting Linux

In an interview, John Carmack suggested that id may stop supporting Linux for their new 'Tech 5 Engine'. So, the next Wolfenstein game may not be coming to Linux.

Space sim shoot-em-ups have been incredibly popular in arcades due to enjoyment of blowing everything up in space while racking up points! Grid Wars 2 is no different. Originally, the game was a clone of the popular *Geometry Wars* on Xbox. Grid Wars brings the space sim to Linux.

The aim of the game is focused around flying a small space craft around the map, shooting at anything on screen. The more objects you

destroy, the more points you gain. Eventually you will be swamped by hundreds of objects and have points in the millions. The primary weapon is sufficient for taking out plenty of ships, plus - with it being unlimited ammo - there will be a lot of just holding down the fire button. Bombs become very useful in situations when you are flooded - they can wipe out everything on the map in one drop. However, use wisely. There is a limited number of bombs. You have three lives, and once you lose all your lives, game over. The game features only one game mode - shoot everything on screen to gain points. After each game, it saves your high scores, which is nice. It's enjoyable trying to beat your best score. Although, it would be nice to have different game modes. Grid wars has different maps to play on; however, the only differences are changes in the background.

The visuals of the game are



simple but affective. Excellent use of colour, and the stylised graphics, suit the game well. The visual effects of explosions and weapon fire all over the map make the game impressive to watch and enjoy. However, I find the sound affects weak. It uses basic sounds for the explosions and weapon fire. The sounds do not complement the graphics and style of the game. This is a shame, since the visual effects are stunning.

The controls are very well suited for the game. It is all controlled using the mouse. Moving the mouse controls the movement of the ship, works like moving a mouse cursor. Fire your primary weapon with left click, and drop a bomb with the right.

Grid Wars is a great game - it is addictive and can be played in quick short bursts. However, unlike other short games I have played, I found that I am playing this game more in the long run. Thanks to its very addictive gameplay, and blowing everything up looks amazing!

The deb installer (technically only for Hardy and Intrepid, but should work in Jaunty) can be found on GetDeb.net at:

[www.getdeb.net/app/GridWars+2](http://www.getdeb.net/app/GridWars+2)

## Score: 7 / 10

### Good:

Amazing Visuals  
Enjoyable gameplay

### Bad:

Poor Sound  
Lack of game modes



**Ed Hewitt**, aka chewit (when playing games), is a keen PC gamer and sometimes enjoys console gaming. He is also on the development team for the Gfire project (Xfire Plugin for Pidgin)





# Q&A

Written by Tommy Alsemgeest

If you have Ubuntu-related questions, email them to: [questions@fullcirclemagazine.org](mailto:questions@fullcirclemagazine.org), and Tommy will answer them in a future issue. Please include as much information as you can about your problem.

**Q** I'm working with Sun Virtual Box, but, every time I start a VM, the window has a transparent background. How can I fix this? I enabled a transparent terminal with "devilspie" before (just in case), but - even without the program running - the problem persists.

**A** It looks like this is a known issue. For now, however, the work-around is to either temporarily or permanently disable compiz effects. If you want to permanently disable it, go to System > Preferences > Appearance, and on to the Visual Effects tab. Click on the "None" radio button, and click "Close". You won't get the pretty effects, but you shouldn't get the transparency either. If, however, you still want the effects, you can disable them while you work in Virtual Box. I find the easiest way to do this is using "Fusion Icon". Install it with:

```
sudo apt-get install fusion-icon
```

and run it from Applications > System Tools > Fusion Icon. Now, when you want to use Virtual Box, right click on the tray icon, Select Window Manager > Metacity (that is, if you are using Ubuntu.) After you have finished with Virtual Box, you can change back to Compiz in the same way that you switched to Metacity.

**Q** I can't install .tar or .tar.gz files! To install these files, I used to extract it to a folder and, changing directory, run the ./configure, (sudo) make, and make install commands in terminal. Every time I run these commands, the terminal replies "bash: ./configure: No such file or directory". As most software comes with those tarball extensions, I can't install any of them.

**A** .tar and .tar.gz files are archives, anything could be inside them. Quite often, in Linux, they will have the source files for the program, which you have to compile before installing, using the commands you have listed. The creator of the archive could put any kind of installer in there, so the best idea is to look for instructions on the site you downloaded it from.

**Q** I coach baseball and I'd like to have the ability to "film" hitting and pitching, and then play it back as slow as frame-by-frame. That way, I can explain to the players exactly what they are doing right and wrong, in each time-segment of their pitching or hitting. Is there a good slow-motion playback program for use with common-format video files, that runs under Ubuntu?

**A** It seems to be possible to do this with VLC. Install it with:

```
sudo apt-get install vlc
```

Open the video, and slow it down by pushing the minus (-) key.

**Q** How can I add the menus that have Applications, Places and System to the bar where the window names show up. In windows, when you open a program they appear in the same panel that has the menu bar, I want to create something like that.

**A** To do this, right click on an empty space on the bottom bar (you may have to move another item to make room), click "Add to panel", scroll down, and select "Menu bar" (or "Main Menu" if you want a more windows-like menu). Now click "Add", then "Close".





# MY DESKTOP

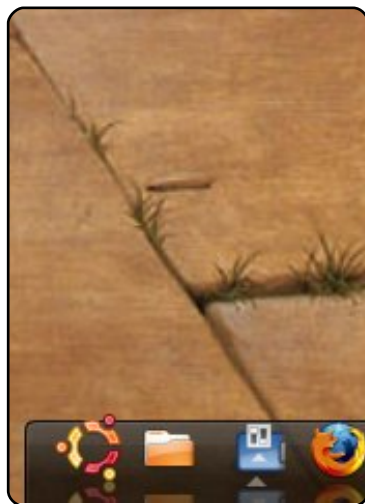
Your chance to show the world your desktop or PC. Email your screenshots and photos to: [misc@fullcirclemagazine.org](mailto:misc@fullcirclemagazine.org) and include a brief paragraph about your desktop, your PC's specs and any other interesting tidbits about your setup.



My name is Wes and I am from New Jersey USA. I have been using Linux for about 3 years now, and can't get enough of it.

I am a desktop support tech, and spend all day working on Windows. So, when I get home, I want something that just works. I can just sit down with my laptop and do what I want - not what the OS wants me to do. I was using Kubuntu, but just made the switch to Ubuntu about two months ago, and have been in bliss.

**Wes** -AKA- Noel Vh.



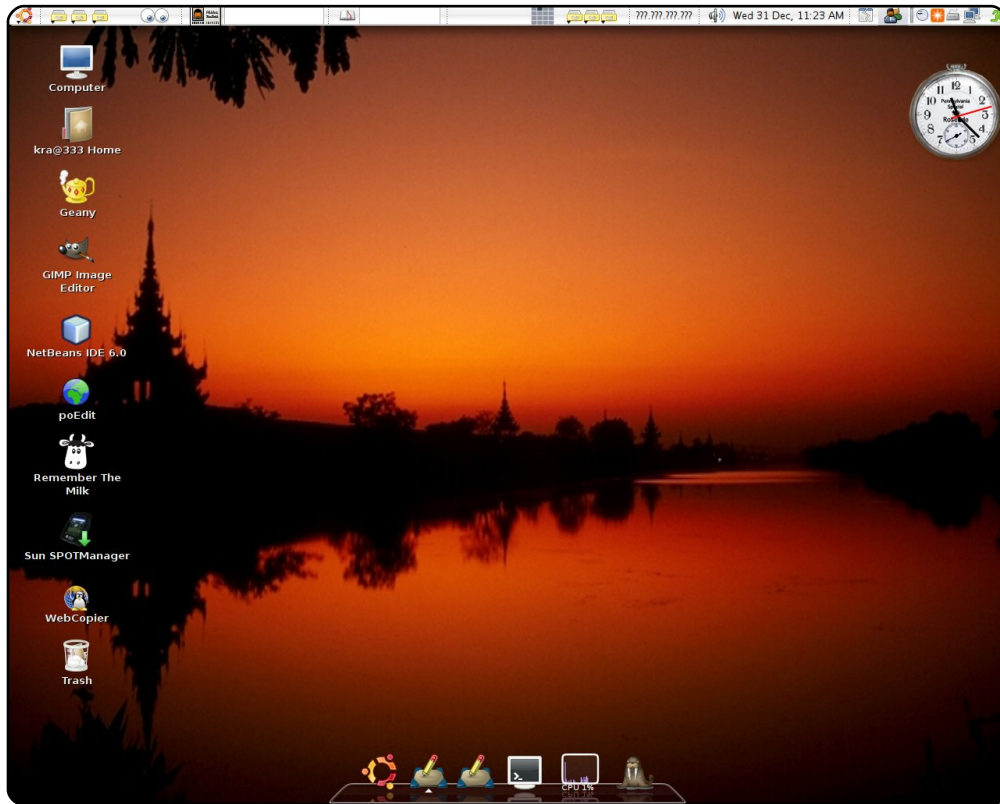
I have been running Ubuntu 8.04 Hardy Heron since about August last year, and various Ubuntu versions for about two years. I dumped Windows about a month after installing 6.06 from a magazine. My present set-up is an AMD Athlon 64 processor and a ASRock motherboard, both of which are, like me, getting a bit long in the tooth.

I prefer an uncluttered desktop, and I don't believe the number of icons Windows users have to wade through to find a program. Using Gimp, I stretched a standard 420x300 graduation to 32x1024 and adjusted the color to use as the background for the bottom panel. Cairo-dock provides the program launcher, but Compiz deals with the rotating cube. I change the wallpaper according to my mood, but most of them come from kde-look.org - this one is called Red Sunrise.

**Brian Cockley**







I am an 18-year-old college student. When I first got my system, it had Windows XP pre-installed. Before that I often used Windows 98. I started using Ubuntu Dapper drake, and have been using Ubuntu for over 18 months.

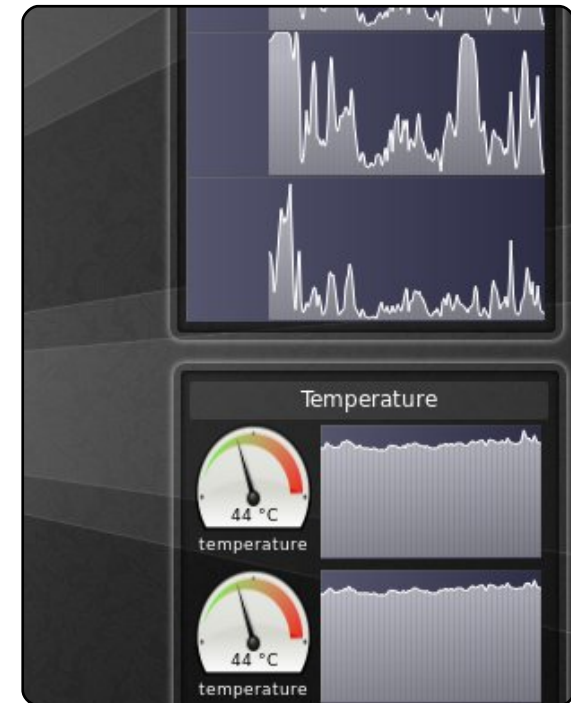
Here's my Ubuntu 7.10 (kernel 2.6.22-14-generic, and Gnome 2.20.0) Desktop. I use a Pentium 4 (2.8 GHz) CPU with 512MB RAM and Original D865Perl Motherboard from Intel. My Ubuntu has the compiz-fusion plug-in. In the picture you see the Emerald theme named "Eternal", and the Icon theme "Gion" - another favourite is "Dropline Neu!". The top panel is configured like the one in Macs. The Awant Doc that you see at the bottom, and my theme, make it look like a real Mac desktop. I have all the tools ranging from programming tools to multimedia applications. I am very happy now: I don't need to format my system often because of viruses or slowdowns, as I often encountered during my Windows 'XP'erience. I am more productive and comfortable with Ubuntu.

**Arun**



This is my personal notebook, an Athlon X2 2.0 GHz x64, 2GB RAM, 160 GB HDD, NVIDIA GeForce 8200M graphic card, and a Realtek Multi-Card Reader. The wallpaper is from [www.guistyles.com](http://www.guistyles.com), and my screen resolution is 1280x800. I've been using Kubuntu since 7.04, and I'm using 8.10 now (Intrepid Ibex), with KDE 4.2, Oxygen icons, and Serenity theme. I use it for studying, and working with maths and programming.

**Ignacio Poggi**

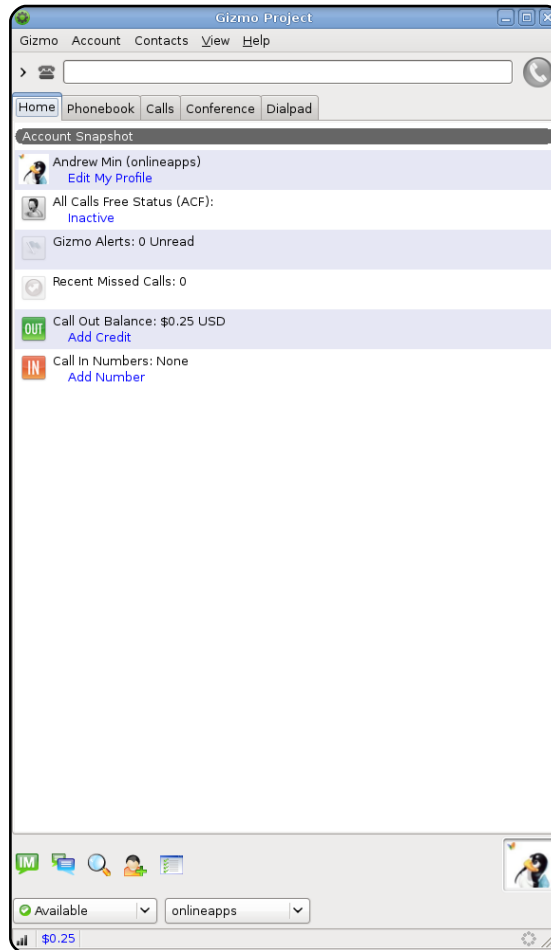


## Gizmo5

<http://gizmo5.com/>

Gizmo5 is one of the oldest SIP programs around. Created by Michael Robertson's SIPphone company, and originally known as the Gizmo Project, Gizmo5 has morphed from an SIP client to the leading SIP host, and challenger to Skype's proprietary service. It sports file transfer, voicemail, and conference calls. For a small fee, you can also receive a phone number and purchase call-out credits (similar to Skype).

Since Gizmo is proprietary, it's not in the Ubuntu repositories. But, you can still install it with an Ubuntu package: use the **libstdc++6** package listed on the download page at <http://url.fullcirclemagazine.org/f6db14>.



## Ekiga

<http://ekiga.org/>

Ekiga, written by Damien Sandras, is the de facto SIP/NetMeeting client installed in Ubuntu. Originally known as GnomeMeeting, it includes a bevy of local network features like support for LDAP, Bonjour/ZeroConf. It also supports standard text, audio, and video conversation tools, plus status messages, an address book, and much more.

Ekiga is installed by default on Ubuntu. To install it on a non-Ubuntu machine, use the **`ekiga`** package.



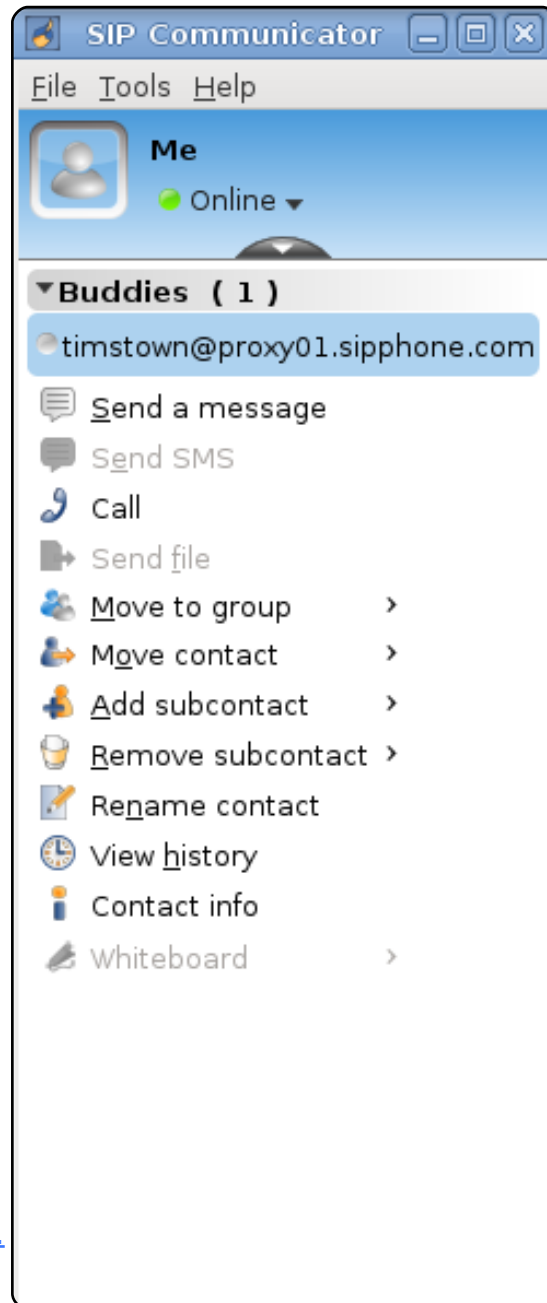


## SIP Communicator

<http://www.sip-communicator.org/>

SIP Communicator wins the (dubious) award for being the only Java-based application on this list. Despite this, it does a heckuva good job fitting in to the Gnome desktop. And it's packed with features: not only does it support standard SIP features, but it also includes support for Jabber, AIM, Yahoo, MSN, and beta support for Facebook. It also has an incredible amount of customizability, including options for media encodings, advanced notifications, and a huge number of plugins.

To install SIP Communicator, use the *Debian* package on the download page at <http://url.fullcirclemagazine.org/70453d>.

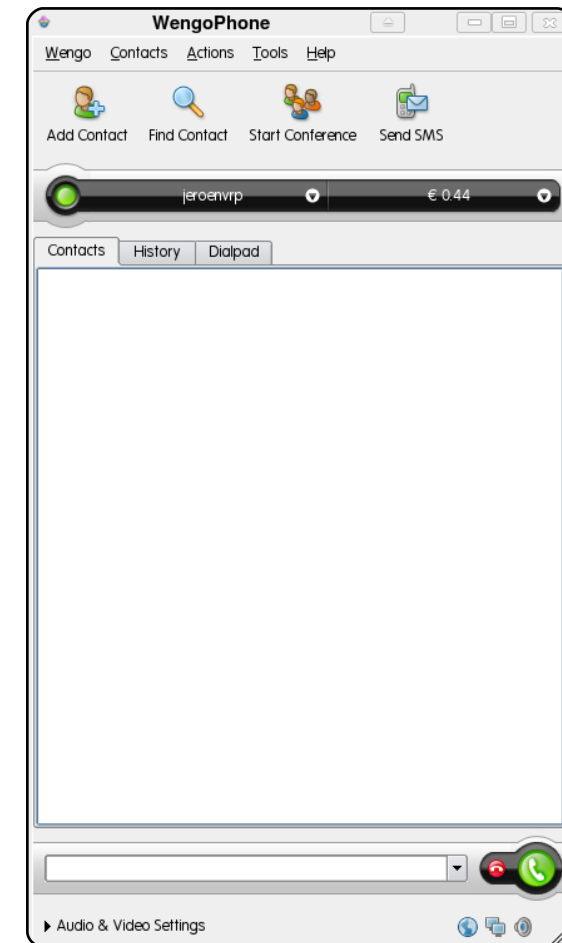


## QuteCom

<http://www.qutecom.org/>

QuteCom, formerly known as WengoPhone, is a powerful Qt-based VoIP client. Like SIP Communicator, it supports multi-protocol chatting, including MSN, AIM, Yahoo, and Jabber. It also supports normal SIP features, including ffmpeg-based video calling, SRTP/AES encryption, and not-so-normal features, including audio emoticons. If you're a KDE user and you like power, QuteCom is an awesome alternative to the Gnome and Java-based programs.

To install QuteCom, use the third-party Launchpad repository at <http://url.fullcirclemagazine.org/1885a6>.

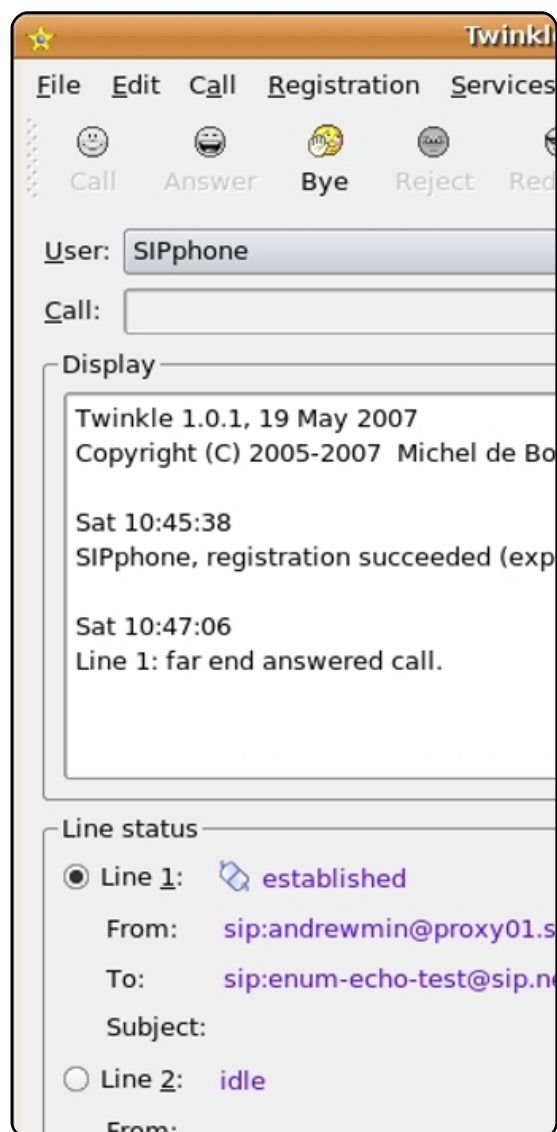


## Twinkle

<http://www.twinklephone.com/>

If you're one of those Kubuntu users who want an alternative to QuteCom, try Twinkle (my client of choice). Written by Michel de Boer, Twinkle packs tons of features, including KAddressBook integration, advanced call handling, highly customizable notifications, encryption, file transfer, and even a command line interface. It also has partial support for presence and a simple address book. All in all, Twinkle is the perfect Qt VoIP tool.

To install Twinkle, use the ``twinkle`` package in the ``universe`` repositories.



**The Ubuntu UK podcast** is presented by members of the United Kingdom's Ubuntu Linux community.

We aim is to provide current, topical information about, and for, Ubuntu Linux users the world over. We cover all aspects of Ubuntu Linux and Free Software and appeal to everyone from the newest user to the oldest coder, from the command line to the latest GUI.

Because the show is produced by the Ubuntu UK community, the podcast is covered by the Ubuntu Code of Conduct and is therefore suitable for all ages.

<http://podcast.ubuntu-uk.org/>



ubuntu uk podcast

Download

Available in MP3/OGG format in Miro, iTunes or listen to it directly on the site.



# HOW TO CONTRIBUTE

We are always looking for new articles to include in Full Circle. For article guidelines, ideas, and for issue translation, please see our wiki:

<http://wiki.ubuntu.com/UbuntuMagazine>

Please email your articles to: [articles@fullcirclemagazine.org](mailto:articles@fullcirclemagazine.org)

If you would like to submit **news**, email it to: [news@fullcirclemagazine.org](mailto:news@fullcirclemagazine.org)

Send your **comments** or Linux experiences to: [letters@fullcirclemagazine.org](mailto:letters@fullcirclemagazine.org)

Hardware/software **reviews** should be sent to: [reviews@fullcirclemagazine.org](mailto:reviews@fullcirclemagazine.org)

**Questions** for Q&A should go to: [questions@fullcirclemagazine.org](mailto:questions@fullcirclemagazine.org)

**Desktop** screens should be emailed to: [misc@fullcirclemagazine.org](mailto:misc@fullcirclemagazine.org)

... or you can visit our **forum** at: [www.fullcirclemagazine.org](http://www.fullcirclemagazine.org)

## FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your Opinions, Desktops and Stories. We also need Reviews (games, apps & hardware), How-To articles (on any K/X/Ubuntu subject) and any questions, or suggestions, you may have.

Send them to: [articles@fullcirclemagazine.org](mailto:articles@fullcirclemagazine.org)

## Full Circle Team



**Editor** - Ronnie Tucker

[ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org)

**Webmaster** - Rob Kerfia

[admin@fullcirclemagazine.org](mailto:admin@fullcirclemagazine.org)

**Comms Mgr** - Robert Clipsham

[mrmonday@fullcirclemagazine.org](mailto:mrmonday@fullcirclemagazine.org)

## Editing & Proofreading

Mike Kennedy

David Haas

Jim Barklow

And our thanks go out to Canonical, the Ubuntu Marketing Team and the many translation teams around the world.

**Deadline for Issue #29:  
Sunday 06th September 2009.**

**Release date for issue #29:  
Friday 25th September 2009.**

